Efficient Domination and Polarity

Dissertation zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

Universität Rostock
Fakultät für Informatik und Elektrotechnik

vorgelegt von
Ragnar Christopher Nevries

urn:nbn:de:gbv:28-diss2014-0183-3

Gutachter:
Prof. Dr. Andreas Brandstädt, Institut für Informatik, Universität Rostock
Prof. Dr. Peter Widmayer, Institut für Theoretische Informatik, ETH Zürich
Prof. Dr. Jing Huang, Mathematics and Statistics, University of Victoria

eingereicht am 5. Februar 2014

verteidigt am 25. Juli 2014

Abstract

This thesis considers Efficient Domination, Efficient Edge Domination, Polarity, and Monopolarity, graph problems that ask for a vertex or edge subset that is a packing and a covering at the same time.

Efficient Domination seeks for an independent vertex subset D such that all other vertices have exactly one neighbor in D. Here, packing means that the vertices of D must not be too close to each other and, in contrast, covering requires that they have to be near to the other vertices. Efficient Edge Domination is the edge version of Efficient Domination. Polarity asks for a vertex subset that induces a complete multipartite graph—the packing aspect—and that contains a vertex of every induced P_3—the covering aspect. Monopolarity is the special case of Polarity where the complete multipartite graph has to be edgeless.

Since all these problems are NP-complete in general, for each problem a lot of effort has been put into separating the graph classes on which the problem remains NP-complete from those that admit an efficient algorithm. This thesis pursues both directions. On the one hand, we introduce a framework for our NP-completeness proofs and use it to sharpen known results for all mentioned problems. On the other hand, we reveal new tractable cases:

Efficient Domination As we figure out that the problem is NP-complete on F-free graphs whenever F is not a linear forest, we clarify the complexity on F-free graphs for linear forests F with at most six vertices, except the P_6. In particular, we show that the problem is efficiently solvable if F has at most five vertices.

Efficient Edge Domination We provide a linear time algorithm for chordal bipartite graphs and an $O(nm)$-time algorithm for hole-free graphs, which solves the open question, whether Efficient Edge Domination is efficiently solvable on weakly chordal graphs.

Monopolarity We formulate a graph property that enables the reduction from Monopolarity to 2-SAT and refine this idea to develop an efficient algorithm for a graph class that contains nearly all known efficiently solvable cases.

Polarity Motivated by our result that Polarity is NP-complete on planar graphs, we introduce an algorithmic framework for subclasses of planar graphs. We show that the framework can be implemented efficiently for hole-free planar and maximal planar graphs.

Acknowledgements

First, I would like to thank my advisor Andreas Brandstädt for his support and for giving me the opportunity to work on this research in a friendly and warm atmosphere. Many thanks also to my colleagues Văn Bằng Lê and Christian Rosenke for having a sympathetic ear for all my scientific and non-scientific questions.

I want to thank my coauthors Andreas, Bang, Christian, and Martin. It was a pleasure to work with you and I learned a lot from it.

Finally, I want to express my gratitude to the patient proofreaders Robert Waltemath and Nicole Nevries. A very special thank to Christian Rosenke for finding the most hidden mistakes and giving me very helpful hints.

Contents

1 Introduction

One fundamental topic in theoretical computer science is the complexity analysis of computational problems, in particular the classification according to the classes P and NP. For practical problems, this classification is indicative for the existence or nonexistence of algorithms that solve the problems exactly and fast enough. Identifying problems that are in P or NP-complete is of high theoretical interest as well, since it is still unknown whether P and NP are really different.

The presented research addresses this topic in the context of algorithmic graph theory. We consider four graph problems, namely EFFICIENT DOMINATION, EFFICIENT EDGE DOMINATION, and POLARITY, as well as MONOPOLARITY, a special case of POLARITY. All these problems ask for a collection of objects that simultaneously is a packing and a covering. Objects of a packing have to be free of "conflicts", while objects of a covering have to "span" all elements of another set. In optimization problems, we seek for largest possible packings and smallest possible coverings.

A prominent example that asks for a collection that is a packing and a covering at the same time is the EXACT COVER problem. It is one of Karp's 21 eminent NP-complete problems published in 1972 [58]. For a given family of subsets $\mathcal{F}$ over a ground set S, it asks whether there is a subset $\mathcal{F}^\star \subseteq \mathcal{F}$ such that every element of S is contained in exactly one member of $\mathcal{F}^\star$. Notice that the packing part seeks for a subset of $\mathcal{F}$ such that its elements are pairwise disjoint and the covering part seeks for a subset of $\mathcal{F}$ such that the union of all its elements is S.

EFFICIENT DOMINATION, introduced in [6], is a special case of EXACT COVER: Given a graph G, it asks if the family $\mathcal{F}$ of the closed neighborhoods of all vertices has a subset $\mathcal{F}^\star$ that exactly covers the vertex set V. In literature, EFFICIENT DOMINATION is also expressed as finding an efficient dominating set, that is, an independent vertex subset D such that all other vertices have exactly one neighbor in D.

EFFICIENT EDGE DOMINATION, introduced in [49], is the edge-version of efficient domination. We seek an edge subset D such that no two of its edges share an endpoint and all other edges share exactly one endpoint with exactly one edge of D.

Polarity, introduced in [91, 92], asks if the vertex set of a graph can be partitioned into a set that induces a complete multipartite graph and a set that induces a disjoint union of complete graphs. The packing aspect of polarity is the demand for a family of independent sets that are pairwise completely connected; the covering aspect requires that every induced path of length 3 has to share a vertex with at least one of these independent sets.

Monopolarity is a well-studied special case of Polarity where the complete multipartite graph has to be edgeless. Hence, we seek for a partition into an edgeless graph and a disjoint union of complete graphs. Notice that efficient edge domination can be formulated as the question whether a graph can be partitioned into an edgeless graph and a 1-regular graph. Since 1-regular graphs are exactly the disjoint unions of complete graphs on two vertices, efficient edge domination is a special case of monopolarity.

Efficient domination, efficient edge domination, polarity, and monopolarity are known to be NP-complete in general and even on very restricted graph classes, that is, when the input graph is supposed to fulfill a certain property. Hence, in the last years, great efforts were made to identify graph classes which allow efficient algorithms. The last three sections of the introduction try to give a comprehensive overview of the achieved results.

We contribute to this research in both directions. On the one hand, we utilize the similarity of the mentioned problems to develop a reduction scheme for NP-completeness proofs. This scheme is directly applicable for efficient domination, efficient edge domination, and monopolarity and we show that the results for monopolarity can be transfered to polarity. We prove that

- efficient domination and efficient edge domination are NP-complete on planar bipartite graphs with maximum degree at most 3 and girth at least g, for every fixed g,
- polarity and monopolarity are NP-complete on planar triangle-free graphs with maximum degree at most 3, and
- polarity and monopolarity are NP-complete on planar graphs with maximum degree at most 3 that contain no induced cycles of a length between 4 and g, for every fixed $g \geq 4$.

Since polarity is closed under taking complements, it also remains NP-complete on the complements of the respective classes, including for example hole-free graphs. Notice that all problems are NP-complete on planar graphs and that the second result for POLARITY/MONOPOLARITY fulfills a weak form of a girth restriction.

On the other hand, we present tractability results for all problems:

EFFICIENT DOMINATION

Using known results, we point out that EFFICIENT DOMINATION is NP-complete on F-free graphs whenever F is not a linear forest. Consequently, we analyze the complexity on F-free graphs for linear forests F. We provide

- a robust $O(nm)$-time algorithm for P_5-free graphs,
- an $O(\min(nm, n^{2.38}))$-time algorithm for P_5-free graphs, and
- a robust $O(nm)$-time algorithm for $(P_4 + P_2)$-free graphs.

All algorithms solve the optimization version of EFFICIENT DOMINATION with vertex weights.

The robust algorithms are achieved by analyzing the structure of the input graph when a fixed vertex is supposed to be part of the solution. In both cases, that is, for P_5-free graphs and for $(P_4 + P_2)$-free graphs, this analysis enables a linear-time algorithm to decide if the input graph admits a solution that contains the fixed vertex. Hence, applying this algorithm for every vertex results in a runtime of $O(nm)$.

The $O(\min(nm, n^{2.38}))$-time result reduces EFFICIENT DOMINATION on P_5-free graphs to MAXIMUM WEIGHT INDEPENDENT SET on cographs.

Together with some simple observations, our results show that EFFICIENT DOMINATION is efficiently solvable on F-free graphs for every linear forest F on at most five vertices. Furthermore, except for the P_6, we clarify the complexity for F-free graphs for every linear forest F on six vertices.

EFFICIENT EDGE DOMINATION

We present

- an $O(n + m)$-time algorithm for chordal bipartite graphs and
- a robust $O(nm)$-time algorithm for hole-free graphs

for the optimization version with edge weights. Since weakly chordal graphs are hole-free, the second result answers a question posed in [17], namely whether EFFICIENT EDGE DOMINATION can be efficiently solved on weakly chordal graphs.

Both algorithms perform a preprocessing on the input graph that yields a K_4-free block graph. For K_4-free block graphs, a linear-time algorithm is known.

Monopolarity

We start by defining a graph class whose monopolar partitions coincide with the satisfying truth assignments of an efficiently computable 2-cnf, that is, a boolean formula in conjunctive normal form with at most two literals per clause. Using preprocessing, we extend this class to a larger graph class that still admits an efficient MONOPOLARITY algorithm. Finally, we use a divide-and-conquer approach, which is based on the block-cutvertex-tree of the input graph, to expand the result to a graph class called locally A_5-$S_{2,2,2}$-defused graphs. This class generalizes well-known graph classes like hole-free graphs, chair-free graphs, and P_5-free graphs and covers nearly all known tractable cases.

Polarity

We develop a framework for POLARITY algorithms on subclasses of planar graphs. Interestingly, the difficult tasks in the framework are highly related to MONOPOLARITY. Hence, based on our MONOPOLARITY results, we show that POLARITY is tractable on maximal planar graphs and hole-free planar graphs. Since we show that POLARITY is NP-complete on planar graphs, especially the latter result corroborates the hypothesis that the complexities of POLARITY and MONOPOLARITY on planar graphs are strongly related.

The thesis is organized as follows. Section 1.1 introduces the used notions and gives some basic facts. The considered graph problems and the state of the art are described in detail in Sections 1.2 to 1.4. Chapters 2 to 4 present the tractability results for EFFICIENT DOMINATION, EFFICIENT EDGE DOMINATION, and POLARITY/MONOPOLARITY respectively. All NP-completeness results are given in Chapter 5, which starts with the reduction framework that is used for all NP-completeness proofs. Chapter 6 concludes the thesis.

All presented results are partially improved versions of results which are published in the article [63], the extended abstracts [8, 12, 64] and the pre-print [13] and which were presented on the "3rd biennial Canadian Discrete and Algorithmic Mathematics Conference" in 2011 (CANADAM'11), "The 22nd International Symposium on Algorithms and Computation" in 2011 (ISAAC'11), and the "38th International Symposium on Mathematical Foundations of Computer Science" in 2013 (MFCS'13). The introduction of each chapter clarifies the differences between the published versions and the versions shown herein.

1.1 Preliminaries

Graph Notions

Let $G = (V, E)$ be a graph with vertex set V and edge set E. We refer to the vertex set and the edge set of a graph G also by $V(G)$ and $E(G)$ respectively. We only consider undirected, simple graphs without loops, that is,

$$E \subseteq \{\{x, y\} \mid x \in V, y \in V, x \neq y\} .$$

For short, we write $xy \in E$ instead of $\{x, y\} \in E$. For an edge $xy \in E$, we say that x and y are the *endpoints* of xy. For a graph $G = (V, E)$ and two vertices x, y with $\{x, y\} \subseteq V$, we say that x and y are *adjacent* or *neighbors* and write $x{-}y$, if $xy \in E$, and we say that x and y are *non-adjacent* or *non-neighbors* and write $x \cdots y$, if $xy \notin E$. For two vertex subsets $X \subseteq V$ and $Y \subseteq V$, we define

$$E(X, Y) := \{xy \mid x \in X, y \in Y, x{-}y\} .$$

We simply write $E(x, Y)$, $E(X, y)$, and $E(x, y)$ for $E(\{x\}, Y)$, $E(X, \{y\})$, and $E(\{x\}, \{y\})$ respectively. For an edge set E, we define $V(E) := \bigcup_{e \in E} e$, that is, the set that contains all endpoints of edges of E.

The *complement* of a graph G, denoted by $\overline{G}$, has the vertex set $V(G)$ and the edge set

$$\overline{E} := \{xy \mid x \in V(G), y \in V(G), x \cdots y, x \neq y\} ,$$

that is, $\overline{G}$ has the edge xy, if and only if G does not have the edge xy.

The *line-graph* $L(G)$ of a graph G has the vertex set $E(G)$ and the edge set

$$\{e_1e_2 \mid e_1 \in E(G), e_2 \in E(G), e_1 \cap e_2 \neq \emptyset, e_1 \neq e_2\} ,$$

that is, the vertices of $L(G)$ are the edges of G and there is an edge in $L(G)$, if and only if the corresponding edges in G share an endpoint.

Two graphs F and G are *isomorphic*, if a bijection $b : V(F) \to V(G)$ exists such that $xy \in E(F)$, if and only if $b(x)b(y) \in E(G)$. A graph F is a *subgraph* of a graph G, if F is isomorphic to a graph F' such that $V(F') \subseteq V(G)$ and $E(F') \subseteq E(G)$. A graph F is an *induced subgraph* of a graph G, if F is isomorphic to a graph F' such that $V(F') \subseteq V(G)$ and $E(F')$ contains every edge $e \in E(G)$ with $e \subseteq V(F')$. This means, that an induced subgraph is a subgraph with the maximal subset of edges. If F is an induced subgraph of G, we also say that F is *induced* in G or that G *contains* F. For a vertex subset $X \subseteq V(G)$, $G[X]$ denotes the induced subgraph of G with the vertex set X. We say that a vertex subset $X \subseteq V(G)$ *induces* the graph F in G, if $G[X]$ is isomorphic to F.

A graph G is said to be $\{F_1, F_2, \ldots, F_k\}$-free, if none of the graphs $F_1, F_2, \ldots, F_k$ is induced in G. For a single graph F, we simply write F-free for $\{F\}$-free.

For a graph $G = (V, E)$, a vertex subset $X \subseteq V$, and an edge subset $M \subseteq E$, we write $G - X$ for $G[V \setminus X]$ and we write $G - M$ for the graph $(V, E \setminus M)$. For a vertex x and an edge e we simply write $G - x$ and $G - e$ instead of $G - \{x\}$ and $G - \{e\}$ respectively.

For a graph $G = (V, E)$ and a vertex subset $U \subseteq V$, we say that a vertex $v \in V$ is *U-universal*, if it is adjacent to every vertex of $U \setminus \{v\}$. For a subgraph H of G, we say that a vertex v is *H-universal*, if v is $V(H)$-universal. A vertex v of a graph G is simply called *universal*, if v is G-universal.

For two graphs F and G with $V(F) \cap V(G) = \emptyset$, vertices $\{x_1, \ldots, x_k\} \subseteq V(F)$, and vertices $\{y_1, \ldots, y_k\} \subseteq V(G)$, the

$$\textit{union} \text{ of } F \text{ and } G \textit{ on } x_1 = y_1,\ x_2 = y_2,\ \ldots,\ x_k = y_k$$

is the graph with the vertex set

$$V(F) \cup V(G) \setminus \{y_1, \ldots, y_k\}$$

that contains

- the edge e for every $e \in E(F)$,
- the edge $x_i v$ for every edge $uv \in E(G)$ with $u = y_i$ and $v \notin \{y_1, \ldots, y_k\}$,
- the edge $x_i x_j$ for every edge $uv \in E(G)$ with $u = y_i$ and $v = y_j$,

and no other edges. Informally one can say that the resulting graph is the union of F and G whereby the vertices $x_1, x_2, \ldots, x_k$ of F are identified with the vertices $y_1, y_2, \ldots, y_k$ of G respectively. So that the union can be used as a commutative operation, in the union graph we refer to the vertex x_i by both names, x_i and y_i.

Notice that this definition of unions only works for vertex disjoint graphs. For two graphs F and G that are not vertex disjoint, we define the union as follows: Construct a graph G' from G by substituting every vertex $x \in V(F) \cap V(G)$ by a new vertex x' and modify the edge relation such that x' has the same neighborhood as x. Clearly, F and G' are vertex disjoint.

Finally, the union of F and G on $x_1 = y_1$, $x_2 = y_2$, $\ldots$, $x_k = y_k$ is defined as the union of F and G' on $x_1 = y_1$, $x_2 = y_2$, $\ldots$, $x_k = y_k$.

For short, we write $(F + G)$ for the union of F and G without identifying any vertices. For a graph G and an integer $m \geq 2$, we inductively define $mG := (G + (m-1)G)$, whereby $1G := G$. That is, mG contains m different copies of the graph G and no edges between vertices of different copies.

The *distance* between two vertices x and y in a graph G, denoted by $\operatorname{dist}(x, y)$, is the smallest number k such that there are $k-1$ vertices $v_1, \ldots, v_{k-1}$ with $x{-}v_1$, $v_1{-}v_2$, $\ldots$, $v_{k-2}{-}v_{k-1}$, and $v_{k-1}{-}y$. If no such k exists, we define $\operatorname{dist}(x, y) = \infty$. The *distance* between two edges xy and $x'y'$ is defined as

$$\operatorname{dist}(xy, x'y') := \min\left\{\operatorname{dist}(x, x'), \operatorname{dist}(x, y'), \operatorname{dist}(y, x'), \operatorname{dist}(y, y')\right\}.$$

The *neighborhood* of a vertex x of a graph G is defined as $N_G(x) := \{y \mid x{-}y\}$. The *closed neighborhood* additionally contains the vertex itself, that is, $N_G[x] := N_G(x) \cup \{x\}$. For a vertex subset $X \subseteq V(G)$, we define its closed neighborhood as $N_G[X] := \bigcup_{x \in X} N_G[x]$ and its neighborhood as $N_G(X) := N_G[X] \setminus X$. Notice that $N_G(X)$ contains no vertex of X, even if X contains adjacent pairs of vertices.

For $d \geq 2$, the *neighborhood in distance* d of a vertex x is defined as $N_G^d(x) := \{y \mid \operatorname{dist}(x, y) = d\}$ and the *closed neighborhood in distance* d as $N_G^d[x] := \{y \mid \operatorname{dist}(x, y) \leq d\}$. For a vertex subset $X \subseteq V(G)$, we define its closed neighborhood in distance d as $N_G^d[X] := \bigcup_{x \in X} N_G^d[x]$ and its neighborhood in distance d as $N_G^d(X) := N_G^d[X] \setminus N_G^{d-1}[X]$, where $N_G^1[X] = N_G[X]$. Notice that a vertex in $N_G^d(X)$ has a distance of at least d to every vertex of X. If it is clear from the context which graph we mean, we just write $N(x)$, $N[x]$, $N(X)$, $N[X]$, $N^d(x)$, $N^d[x]$, $N^d(X)$, and $N^d[X]$ respectively.

The *degree* of a vertex x in a graph G is defined as $\deg_G(x) := |N_G(x)|$. Again, we simply write $\deg(x)$ if G is clear from the context. Sometimes, for a vertex subset $Y \subseteq V(G)$, we write $\deg_Y(x)$ for $|N_G(x) \cap Y|$. Vertices of degree 0 are called *isolated*, vertices of degree 1 are called *pending vertices* and edges with one endpoint of degree 1 are called *pending edges*.

A graph is *connected*, if every pair of vertices has finite distance. A *connected component* of a graph is an inclusion-maximal induced subgraph that is connected. Consequently, a graph is *co-connected*, if its complement is connected and a *co-connected component* is a connected component of the complement of the graph. A vertex c of a graph G is called *cutvertex*, if $G - c$ has more connected components than G. A graph is *biconnected*, if it contains no cutvertex. A *biconnected component* of a graph, also called *block*, is an inclusion-maximal induced subgraph that is biconnected. We say that a block is *trivial*, if it contains at most two vertices, otherwise it is called *non-trivial*. The following fact is well known, but we give a proof for the sake of completeness:

Fact 1. *Let $G = (V, E)$ be a graph and let $B_1 = (V_1, E_1), \dots, B_k = (V_k, E_k)$ be the blocks of G. For the accumulated size of the blocks we have*

$$|V_1| + \dots + |V_k| \le 2|V| \text{ and } |E_1| + \dots + |E_k| = |E|.$$

Proof. If G is not connected, the bounds clearly hold for G if they hold on each connected component. Hence, we can assume that G is connected.

Every edge of G is in exactly one block because if there is an edge in two blocks, B_i and B_j, these blocks share two vertices and, hence, $V(B_i) \cup V(B_j)$ induces a biconnected subgraph of G—this is a contradiction to the maximality of B_i and B_j.

We show the bound on the vertex sum of the blocks by induction on the number of blocks. Clearly, if G is biconnected, that is, it has only one block, the bound holds. Hence, assume that G is a graph with blocks $B_1 = (V_1, E_1), \dots, B_k = (V_k, E_k)$, $k > 1$. Assume that for every graph with at most $k - 1$ blocks the bound holds. Since G is not biconnected, there is a block with just one cutvertex. Assume without loss of generality that B_k is such a block. Let $G' = (V', E') := G[V \setminus V_k \cup \{v\}]$. The blocks of G' are $B_1, \dots, B_{k-1}$. Hence, we can apply the induction hypothesis

$$|V_1| + \dots + |V_{k-1}| \le 2|V'|,$$

and, since $|V_k| \ge 2$, we get

$$\begin{aligned} & |V_1| + \dots + |V_{k-1}| + |V_k| \\ \le\ & 2|V'| + |V_k| \\ =\ & 2\left(|V| - |V_k| + 1\right) + |V_k| \\ =\ & 2|V| - |V_k| + 2 \le 2|V|. \end{aligned}$$

□

For a graph G, a *vertex weight function* is a function $\omega : V(G) \to \mathbb{R}^+$ and an *edge weight function* is a function $\omega : E(G) \to \mathbb{R}^+$. For short, we say that G has vertex weights ω or G has edge weights ω, if ω is a vertex weight function of G or an edge weight function of G respectively. For any function $\omega : X \to \mathbb{R}^+$ and a subset $X' \subseteq X$, we define $\omega(X') := \sum_{x \in X'} \omega(x)$.

Graphs and Graph Classes

The graph P_k, also called *chordless k-path* or *chordless path of length $k-1$*, is the graph with vertex set $\{x_1, \ldots, x_k\}$ and edge set $\{x_i x_{i+1} \mid 1 \leq i < k\}$. Analogously, for $k \geq 3$, the graph C_k, also called *chordless k-cycle* or *chordless cycle of length k*, is the graph P_k with the additional edge $x_1 x_k$. The chordless 3-cycle is also called *triangle* and every chordless k-cycle with $k \geq 5$ is also called *hole*.

A *k-path* or *path of length $k-1$*, denoted by $x_1 - x_2 - \ldots - x_k$, is a graph with vertex set $\{x_1, \ldots, x_k\}$ and at least the edges $\{x_i x_{i+1} \mid 1 \leq i < k\}$. Analogously, for $k \geq 3$, the *k-cycle* or *cycle of length k* is a k-path with the additional edge $x_1 x_k$ and denoted by $x_1 - x_2 - \ldots - x_k - x_1$. Notice that the chordless k-path is a k-path and that the chordless k-cycle is a k-cycle, but not vice versa. Furthermore, notice that every k-cycle is a k-path and that a k-path can also be a k-cycle.

For every $k \geq 1$, the k-path is also called *path* and, for every $k \geq 3$, the k-cycle is also called *cycle*. For a graph G that contains a path $x_1 - x_2 - \ldots - x_k$ or a cycle $x_1 - x_2 - \ldots - x_k - x_1$, we say that $x_1 - x_2 - \ldots - x_k$ or $x_1 - x_2 - \ldots - x_k - x_1$ is *induced* in G, if $\{x_1, \ldots, x_k\}$ induces a P_k or a C_k in G respectively.

For two paths $P = x_1 - x_2 - \ldots - x_{k'}$ and $Q = x_{k'} - x_{k'+1} - \ldots - x_k$, we define $P.Q := x_1 - x_2 - \ldots - x_k$.

A *shortest path* between two vertices x and y is a path $x - \ldots - y$ of G of length $\operatorname{dist}(x, y)$. Clearly, if $\operatorname{dist}(x, y) = \infty$, no shortest path between x and y exists. A graph is called *acyclic*, if it contains no cycle. A graph has *girth* g, if every induced cycle has length at least g. For an acyclic graph, the girth is infinite. The *maximum degree* of a graph G is the maximum of the degrees of all vertices of G. A graph is called *r-regular*, if every vertex has degree r. Edge sets that induce 1-regular graphs are called *induced matchings*.

A *complete* graph contains all possible edges. We call a vertex subset X of a graph G *independent*, if $G[X]$ is edgeless, and we call it a *clique*, if $G[X]$ is complete. A graph G is called *k-partite*, if there is a partition $V(G) = V_1 \cup \cdots \cup V_k$ such that, for every $i \in \{1, \ldots, k\}$, V_i is an independent set in G. We denote by $G = (V_1 \cup \cdots \cup V_k, E)$ a k-partite graph with independent sets $V_1, \ldots, V_k$. The 2-partite graphs are also called *bipartite*. A graph G is *complete k-partite*, if there is a partition $V(G) = V_1 \cup \cdots \cup V_k$ such that, for every $i \in \{1, \ldots, k\}$, V_i is an independent set in G and, for every $j \in \{1, \ldots, k\}, i \neq j$, there are all possible edges in $E(G)(V_i, V_j)$. A graph is *(complete) multipartite*, if it is (complete) k-partite for some $k \geq 1$. Notice

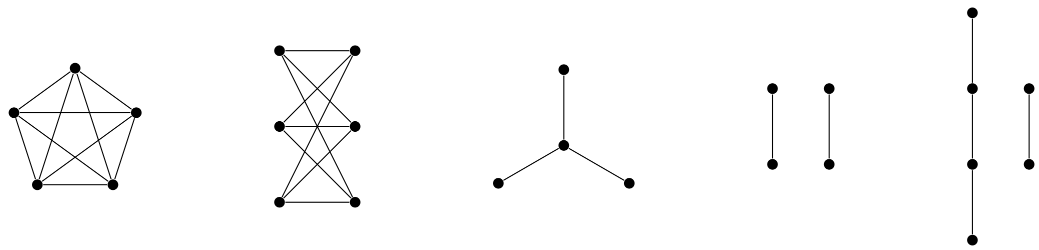

Figure 1.1: The graphs K_5, $K_{3,3}$, claw ($K_{1,3}$), $2P_2$, and $(P_4 + P_2)$ (from left to right).

that a graph is complete multipartite, if and only if it is $\overline{P_3}$-free.

The complete graph on k vertices is denoted by K_k. The complete bipartite graph $G = (X \cup Y, R)$ with $|X| = i$ and $|Y| = j$ is denoted by $K_{i,j}$. The graph $K_{1,3}$ is also called *claw*. A *linear forest* is an acyclic claw-free graph, that is, the union of chordless paths. Figure 1.1 depicts the graphs K_5, $K_{3,3}$, claw and, as examples for linear forests, the graphs $2P_2$ and $(P_4 + P_2)$.

A graph is called *line-graph*, if it is isomorphic to $L(G)$ for some graph G. A graph is *chordal*, if every induced chordless cycle has length 3, that is, chordal graphs are the $\{C_4, C_5, \ldots\}$-free graphs. A graph is *chordal bipartite*, if every induced chordless cycle has length 4. Notice that this name is slightly misleading because chordal bipartite graphs are bipartite but not chordal.

A graph is *planar*, if it can be embedded in the plane, that is, the vertices are points and the edges are curves connecting the corresponding points, such that no edge crossings appear. The *outer face* of an embedding of a planar graph is the face of the drawing with infinite area. For a graph G and an edge xy of G, *contracting* xy yields the graph G' that results from the graph $G - y$ by adding a minimal set of edges such that $N_G(y) \subset N_{G'}(x)$. A graph F is a *minor* of a graph G, if F can be constructed from G by repeatedly deleting vertices or edges or contracting edges. By the theorems of Kuratowski and Wagner, the planar graphs are exactly the $\{K_{3,3}, K_5\}$-minor-free graphs. This implies that neither the K_5 nor the $K_{3,3}$ can be a subgraph of a planar graph.

A graph class $\mathcal{C}$ is called *hereditary*, if for every graph $G \in \mathcal{C}$ and every induced subgraph F of G, we have $F \in \mathcal{C}$. Similarly, a graph class is called *additive*, if for every two graphs $F \in \mathcal{C}$ and $G \in \mathcal{C}$, we have $(F + G) \in \mathcal{C}$.

For graph classes that are not defined here, we refer to the comprehensive survey [9].

Efficient Dominating Sets and Polar Partitions

Let $G = (V, E)$ be a graph. A vertex subset X is *efficient*, if every pair of vertices of X has a distance of at least 3, that is, X is independent and every vertex of $V \setminus X$ has at most one neighbor in X. An edge subset M is *efficient*, if every pair of edges of M has a distance of at least 2, that is, no two edges of M share an endpoint and every edge of $E \setminus M$ shares at most one endpoint with an edge of M.

A vertex subset X *covers* all edges of E that have at least one endpoint in X. An edge subset M *covers* all vertices of $V(M)$.

A vertex subset X *dominates* all vertices in $N[X]$, that is, it dominates itself and all its neighbors. An edge subset M *dominates* all edges of M and all edges of E that share an endpoint with an edge of M. Consequently, X is called *dominating*, if it dominates all vertices of G and M is called *edge dominating*, if it dominates all edges of G.

A vertex subset X is an *efficient dominating set*, if it is efficient and dominating. An edge subset M is an *efficient edge dominating set*, if it is efficient and edge dominating. Notice that an edge subset M is an efficient edge dominating set in G, if and only if M is an efficient dominating set in the line-graph $L(G)$.

A graph G is called *efficiently dominatable*, if it has an efficient dominating set, and it is called *efficiently edge dominatable*, if it has an efficient edge dominating set.

A vertex partition $V = A \cup B$, denoted by (A, B), is *polar*, if $G[A]$ is a complete multipartite graph and $G[B]$ is P_3-free. A polar partition (A, B) is *monopolar*, if $G[A]$ is edgeless, and *unipolar*, if $G[A]$ is complete. Notice that the edges in $E(A, B)$ are arbitrary. For a polar, monopolar, or unipolar partition (A, B), we say that the vertices of A are `amber` *colored* and the vertices of B are `blue` *colored*.

A graph G is *polar*, *monopolar*, or *unipolar*, if it admits a polar, monopolar, or unipolar partition respectively. Notice that a graph is complete multipartite, if and only if it is $\overline{P_3}$-free. This immediately implies that the complement of a polar graph is polar. This is not necessarily true for monopolar or unipolar graphs.

A pair of two disjoint vertex subsets, that is, $A', B' \subseteq V$ and $A' \cap B' = \emptyset$, denoted by (A', B'), is called a *precoloring* of G. For a precoloring (A', B'), we say that the vertices of A' are `amber` *precolored* and the vertices of B' are `blue` *precolored*.

For a precoloring (A', B'), a monopolar partition (A, B) is a *monopolar* *extension* of (A', B') and G, if $A' \subseteq A$ and $B' \subseteq B$; we also say that (A, B) is an (A', B')-*monopolar extension* of G. A graph is (A', B')-*monopolar extendable*, if it admits a vertex partition that is an (A', B')-monopolar extension.

Computational Complexity and Robust Algorithms

In classical computational complexity theory, the runtime of an algorithm is measured as the number of steps that a certain computation model needs to execute that algorithm. The runtime is normally given as a function of the encoding length of the input. In the presented algorithms, the input often contains a graph, hence, we define $|G|$ as the *encoding length* of a graph $G = (V, E)$. It is common to assume that the graphs are explicitly given in the input, so we can suppose that $|G| \geq |V| + |E|$, no matter which computation model is used. To entirely abstract from the computation model, for graph algorithms, the runtime often is given in respect to the number of vertices and edges of the input graph.

Another measure that highly depends on the computation model is the time that is needed for basic operations like integer comparison or basic arithmetic functions like addition or multiplication. We assume that these basic operations need constant time, thus, we give the runtime as the number of basic operations. This is also known as arithmetic complexity and widely used for the time complexity analysis of algorithms.

Some of the presented algorithms are *robust*. An algorithm `A` is said to work on a graph class $\mathcal{C}$ in a *robust way*, if `A` works on every input graph G and either returns a correct output or the statement that G is not in $\mathcal{C}$. Notice that it is possible that `A` correctly terminates on graphs that are not in $\mathcal{C}$. Algorithms of this kind are of special interest if their runtime is less than the runtime for deciding if a graph is in $\mathcal{C}$. For further information we refer to [86].

1.2 Efficient Domination

In graph theory, the concept of domination is a very important and well-studied topic. A decent introduction would go beyond the scope of this thesis. We therefore refer the reader to the textbooks [51, 52] and the survey [55] as a good starting point.

Efficient dominating sets first appear as *perfect* 1-*codes* in an article of Biggs about perfect codes in graphs [6]. This work is motivated by the concept

Figure 1.2: The graph to the left (known as 4-*wheel* or W_4) is efficiently dominatable by the circled vertex. The graph to the right (known as C_4) is not efficiently dominatable. Notice that the C_4 is an induced subgraph of the W_4.

of error correcting codes and translates it to the domain of graphs. It gives a criterion for the existence of perfect e-codes in distance-transitive graphs. A perfect e-code is a vertex set X such that $\bigcup_{x \in X} N^e[x]$ is a partition of the vertex set.

Later, the question whether a graph admits a perfect 1-code was considered as EFFICIENT DOMINATION:

EFFICIENT DOMINATION	
`Input:`	A graph $G = (V, E)$.
`Question:`	Is there a vertex subset $D \subseteq V$ such that D is an efficient dominating set of G?

The problem can be formulated in several ways. For example, as mentioned in the introduction, it coincides with the question whether the family of the closed neighborhoods of all vertices, that is, $\{N[v] \mid v \in V\}$, admits an exact cover of V. In other words, can the vertex set V be partitioned into sets $V_1 \cup \cdots \cup V_k$ such that every V_i equals $N[v]$ for some vertex $v \in V$? Furthermore, EFFICIENT DOMINATION can be expressed in terms of generalized dominating sets, see for example [54].

Notice that EFFICIENT DOMINATION can be expressed in monadic second order logic without edge set quantification. By the way, this is also true for EFFICIENT EDGE DOMINATION, MONOPOLARITY, and POLARITY.

The definition as decision problem is justified because there are even pretty simple graphs that have no efficient dominating set. For example, Figure 1.2 shows two graphs: The W_4 is efficiently dominatable by simply taking $\{x\}$ as efficient dominating set. The C_4 admits no efficient dominating set because whenever a vertex is chosen, say v, then its non-neighbor, z, cannot be dominated without violating efficiency.

Since the C_4 is an induced subgraph of the W_4, the figure also demonstrates that efficiently dominatable graphs are not hereditary. However, they are additive: For an efficient dominating set D_F of a graph F and an efficient dominating set D_G of a graph G, one can easily check that $D_F \cup D_G$ is an efficient dominating set of $(F+G)$.

Besides the decision version, one can also formulate an optimization version:

MINIMUM/MAXIMUM (WEIGHT) EFFICIENT DOMINATION	
`Input:`	A graph $G = (V, E)$ (with vertex weights ω) and a number $k \in \mathbb{R}^+$.
`Question:`	Is there a vertex subset $D \subseteq V$ such that D is an efficient dominating set of G and $\|D\|$ is at most/at least k ($\omega(D)$ is at most/at least k)?

All algorithms for EFFICIENT DOMINATION that are presented in this thesis solve the weighted minimization problem. In fact, most of them can easily be modified to solve the maximization version and to output an efficient dominating set of minimum/maximum weight, if it exists.

In 1988, EFFICIENT DOMINATION was shown to be NP-complete by Bange, Barkauskas, and Slater [4]. Several results for restricted graph classes followed: EFFICIENT DOMINATION remains NP-complete on

- planar graphs with maximum degree at most 3 [45],
- bipartite graphs [94],
- chordal graphs [94],
- line-graphs of bipartite graphs [72],
- planar bipartite graphs [73],
- chordal bipartite graphs [73], and
- chordal unipolar graphs (by the reduction given in [94], noted in [39]),

ordered by the year of publication.

In Section 5.2, we show that EFFICIENT DOMINATION is NP-complete on planar bipartite graphs with maximum degree at most 3 and girth at least g, for every fixed g.

In [4], Bange et al. show that EFFICIENT DOMINATION can be solved on trees in linear time. Since then a lot of tractability results were found. The following list is ordered by the year of publication. MINIMUM WEIGHT EFFICIENT DOMINATION is solvable

- in linear or polynomial time on graphs with bounded tree-width [1, 31, 33],
- in time $O(n+m)$ on split graphs [20],
- in time $O(n+m)$ on interval graphs [21],
- in time $O(nm+n^2)$ on circular-arc graphs [21],
- in time $O(n^{2.37})$ on co-comparability graphs [19],
- in linear time on block graphs [94],
- in time $O(n+\overline{m})$ on permutation graphs [65],
- in time $O(n \log\log n+\overline{m})$ on trapezoid graphs [65],
- in time $O(n \log n)$ on trapezoid graphs [67],
- in linear or polynomial time on graphs with bounded clique-width [32],
- in time $O(n)$ on bipartite permutation graphs [73], and
- in time $O(n)$ on distance-hereditary graphs [73],

where n is number of vertices and m the number of edges of the input graph and $\overline{m}$ is the number of edges of the complement of the input graph. Besides these results for the weighted minimization version, there are the following recent results: In [76], Milanič characterizes the hereditary efficiently dominatable graphs as the $\{\text{bull}, \text{fork}, C_{3k+1}, C_{3k+2}\}$-free graphs and gives a polynomial-time recognition algorithm for this class that outputs an efficient dominating set, if possible. Brandstädt et al. [10, 11] show that the decision version is polynomial-time solvable on asteroidal triple-free graphs, interval bigraphs, and dually chordal graphs. Actually, in the manuscript [7] it is shown that the weighted optimization versions are also polynomial-time solvable on these graph classes.

In Chapter 2, we consider graph classes that are characterized by a single forbidden subgraph. We argue that EFFICIENT DOMINATION is NP-complete on such classes, whenever the forbidden induced subgraph is not a linear forest. For that reason, this thesis starts a systematic analysis of graph classes without induced linear forests and contributes the following results: MINIMUM WEIGHT EFFICIENT DOMINATION is solvable

- in time $O(nm)$ on P_5-free graphs in a robust way,

- in time $O(\min\{nm, n^{2.37}\})$ on P_5-free graphs, and
- in time $O(nm)$ on $(P_4 + P_2)$-free graphs in a robust way.

Together with trivial results and corollaries of other results, this shows that MINIMUM WEIGHT EFFICIENT DOMINATION is efficiently solvable on F-free graphs for every linear forest F on at most five vertices. Furthermore, when considering F-free graphs for linear forests F on six vertices, $F = P_6$ is the only case where the complexity is still open.

1.3 Efficient Edge Domination

Efficient edge dominating sets were introduced in [49]. The corresponding EFFICIENT EDGE DOMINATION problem is motivated by resource allocation problems in parallel processing systems [68], encoding theory, and network routing problems.

EFFICIENT EDGE DOMINATION	
`Input:`	A graph $G = (V, E)$.
`Question:`	Is there an edge subset $D \subseteq E$ such that D is an efficient edge dominating set of G?

Since, for a graph $G = (V, E)$, an edge subset M is efficient edge dominating, if and only if it is an efficient dominating set in the line-graph of G, EFFICIENT EDGE DOMINATION can be expressed as EFFICIENT DOMINATION on line graphs.

In literature, the problem is also formulated as follows: Can the vertex set of a given graph G be partitioned into two sets A and B such that $G[A]$ is edgeless and $G[B]$ is 1-regular? This is equivalent to our definition because one can easily check that D is an efficient edge dominating set of a graph G, if and only if $G - V(D)$ is edgeless and $G[V(D)]$ is 1-regular.

Perfect matching are also known as induced matchings and, in literature, EFFICIENT EDGE DOMINATION is therefore also called DOMINATING INDUCED MATCHING. This gives a relation to the MAXIMUM INDUCED MATCHING problem, as studied in [16], for example.

Furthermore, this formulation correlates EFFICIENT EDGE DOMINATION to MONOPOLARITY, which asks for a partition of the vertex set such that $G[A]$ is edgeless and $G[B]$ is P_3-free. Since 1-regular graphs are P_3-free, EFFICIENT EDGE DOMINATION is a specialization of MONOPOLARITY.

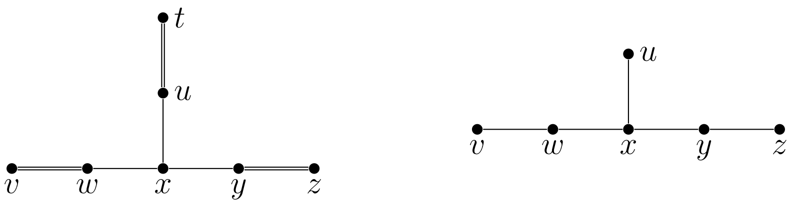

Figure 1.3: The graph to the left (namely the graph $S_{2,2,2}$) is efficiently edge dominatable by taking the edges drawn with double lines. The graph to the right (namely the graph $S_{1,2,2}$) is not efficiently edge dominatable. Notice that the $S_{1,2,2}$ is an induced subgraph of the $S_{2,2,2}$.

Similar to EFFICIENT DOMINATION, even very simple graphs have no efficient edge dominating set. For example, Figure 1.3 shows two graphs, the $S_{2,2,2}$ and the $S_{1,2,2}$: Taking the edges $\{tu, vw, yz\}$ of the $S_{2,2,2}$ yields an efficient edge dominating set. The $S_{1,2,2}$ is not efficiently edge dominatable because, if we choose any edge, then at least one of the edges $\{ux, vw, yz\}$ cannot be dominated without violating efficiency.

The $S_{2,1,1}$ is an induced subgraph of the $S_{2,2,2}$, hence, efficiently edge dominatable graphs are not hereditary. However, they are additive: Consider two vertex disjoint graphs F and H, let D_F be an efficient edge dominating set of F, and let D_H be an efficient edge dominating set of H. One can easily check that $D_F \cup D_H$ is an efficient edge dominating set of $(F + G)$.

The optimization versions of EFFICIENT EDGE DOMINATION are:

MINIMUM/MAXIMUM (WEIGHT) EFFICIENT EDGE DOMINATION	
`Input:`	A graph $G = (V, E)$ (with edge weights ω) and a number $k \in \mathbb{R}^+$.
`Question:`	Is there an edge subset $D \subseteq E$ such that D is an efficient edge dominating set of G and $\lvert D\rvert$ is at most/at least k ($\omega(D)$ is at most/at least k)?

EFFICIENT EDGE DOMINATION was shown to be NP-complete in general in the introductory work [49]. Sharper NP-completeness results, ordered by the year of publication, followed:

- bipartite graphs [71],
- planar bipartite graphs [70],
- r-regular graphs for every $r \geq 3$ [16], and

- bipartite graphs of maximum degree at most 3 [17, 18].

In particular, the latter result shows NP-completeness for the class of $\{C_3, \ldots, C_k, H_1, \ldots, H_k\}$-free bipartite graphs of maximum degree at most 3 for every k, where H_i is the graph consisting of two P_3 whose vertices of degree 2 are connected by a chordless path of length i.

In Section 5.3, we show that EFFICIENT EDGE DOMINATION remains NP-complete on planar bipartite graphs with maximum degree at most 3 and girth at least g, for every fixed g.

On the other hand, there is a variety of tractability results, some of them dealing with the decision version, others dealing with the weighted optimization version. The following list is ordered by the year of publication and we indicate in parentheses which version is considered. EFFICIENT EDGE DOMINATION is solvable

- in linear or polynomial time on graphs with bounded tree-width [1, 31, 33] (decision),
- in time $O(n)$ on generalized series-parallel graphs [49] (decision),
- in time $O(n+m)$ on bipartite permutation graphs [72] (weighted optimization),
- in linear or polynomial time on graphs with bounded clique-width [32] (decision),
- in time $O(n+m)$ on generalized series-parallel graphs [70] (weighted optimization),
- in polynomial time on convex graphs [17, 59] (decision),
- in time $O(n^2)$ on claw-free graphs [17, 18] (decision),
- in polynomial time on $\{H_i, H_{i+1}, \ldots\}$-free graphs with maximum degree at most d for every i and d [17] (decision),
- in time $O(n+m)$ on P_7-free graphs [14] (weighted optimization),
- in time $O(n+m)$ on dually chordal graphs [10, 11] (decision),
- in polynomial time on interval-filament graphs [10, 11] (decision),
- in polynomial time on asteroidal triple-free graphs [10, 11] (decision),

- in polynomial time on weakly chordal graphs [10, 11] (decision), and
- in polynomial time on $S_{1,2,3}$-free graphs [60] (decision),

where n and m are the number of vertices and edges of the input graph respectively.

In Chapter 3, we show that MINIMUM WEIGHT EFFICIENT EDGE DOMINATION is solvable

- in time $O(n+m)$ on chordal bipartite graphs and
- in time $O(nm)$ on hole-free graphs in a robust way.

The result for hole-free graphs answers the question whether EFFICIENT EDGE DOMINATION is efficiently solvable on weakly chordal graphs posed in [17]. This was of interest because weakly chordal graphs generalize chordal graphs and convex graphs. In fact, the question is also answered in [10, 11] by showing that EFFICIENT EDGE DOMINATION on weakly chordal graphs can be reduced to MAXIMUM INDEPENDENT SET on weakly chordal graphs and, hence, can be solved in time $O(n^4)$. Moreover, it is argued that every efficiently edge dominatable hole-free graph is weakly chordal, so the result also solves EFFICIENT EDGE DOMINATION on hole-free graphs. But, on the one hand, our result was published earlier in [8] and, on the other hand, the algorithm given in this thesis is robust and has a better time complexity, even better than recognizing hole-free or weakly-chordal graphs.

1.4 Polarity

Polar graphs are a natural generalization of bipartite graphs and split graphs introduced by Tyshkevich and Chernyak in 1985 [91, 92]. By definition, they properly contain all monopolar graphs and all unipolar graphs. Since the recognition of polar graphs is NP-complete, even on very restricted graph classes, it is interesting to analyze the complexity of recognizing monopolar graphs and unipolar graphs. Therefore, we define the recognition problems as follows:

POLARITY (MONOPOLARITY, UNIPOLARITY)	
`Input:`	A graph $G = (V, E)$.
`Question:`	Is there a polar (monopolar, unipolar) partition (A, B) of the vertex set V?

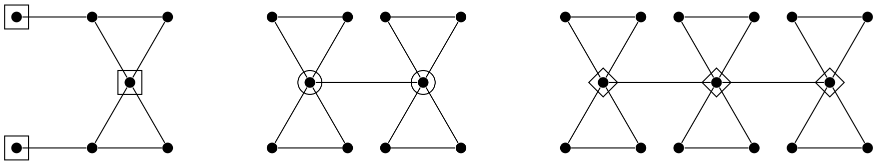

Figure 1.4: The leftmost graph (F_1) is monopolar by coloring the squared vertices `amber`, the graph at center (F_2) is unipolar by coloring the circled vertices `amber`, and the rightmost graph (F_3) is polar by coloring the vertices in diamonds `amber`.

Polar partitions are special sparse-dense partitions, thus, POLARITY is a special case of the MATRIX PARTITION problem [43, 44]. Furthermore, polar partitions can be regarded as generalized colorings [15].

In literature, the set of `amber` colored vertices of a monopolar partition is also called *independent P_3-transversal*. This is a reasonable way of thinking about monopolar partitions: Find an independent set that contains at least on vertex of every induced P_3.

To substantiate the fact that monopolar graphs and unipolar graphs are proper subsets of polar graphs, Figure 1.4 shows three graphs, F_1, F_2, and F_3: The graph F_1 is monopolar by coloring the pending vertices and the vertex of degree 4 `amber` and the other vertices `blue`. It is easy to verify that this graph admits no unipolar partition. The graph F_2 is unipolar by taking the two vertices of degree 5 into the set A and the other vertices into the set B. Again, it is easy to verify that this graph admits no monopolar partition. Finally, the graph F_3 has neither a monopolar nor a unipolar partition, but it is polar by coloring the vertices of degree 2 `blue` and the other vertices `amber`.

It is also possible that a graph is monopolar and additionally admits polar partitions that are not monopolar. Figure 1.5 shows an example for such a graph. The partition (A, B) with $A = \{r, u, v, y\}$ and $B = \{q, s, t, w, x, z\}$ is polar: The induced subgraph $G[B]$ consists of the two disjoint triangles $q–t–x–q$ and $s–w–z–s$, which is clearly P_3-free. The induced subgraph $G[A]$ is a complete multipartite graph because one can partition its vertices into three completely connected independent sets, namely $\{r, y\}$, $\{u\}$, and $\{v\}$. But this partition is not monopolar, although the graph also admits the monopolar partition (A, B) with the independent set $A = \{r, t, w, y\}$ and $B = \{q, s, u, v, x, z\}$, which induces a P_3-free graph.

In contrast to efficiently dominatable and efficiently edge dominatable graphs, the classes of polar, monopolar, and unipolar graphs are hereditary.

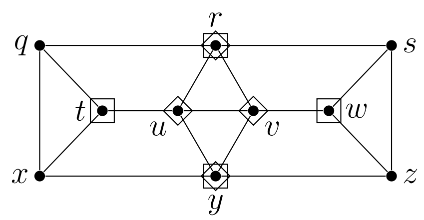

Figure 1.5: This graph is monopolar by coloring the squared vertices `amber`. Coloring the vertices in diamonds `amber` yields a polar partition that is not monopolar.

This can easily be realized by the fact that complete multipartite graphs, complete graphs, edgeless graphs, and P_3-free graphs are hereditary graph classes. But polar graphs and unipolar graphs are not additive. As an example, consider the graph F_2 of Figure 1.4. This graph is unipolar and, hence, polar, but it admits no monopolar partition. Now consider $G = 2F_2$. Since monopolar graphs are hereditary, G cannot admit a monopolar partition. Since complete graphs and complete multipartite graphs are connected, it is impossible to choose vertices from both connected components of G into the set A of a polar partition. Hence, since both connected components contain induced 3-paths, G is not polar.

Even in published articles this fact is overlooked. For example, [35] shows that POLARITY is efficiently solvable on connected permutation graphs and then, it is stated that this implies efficient recognition of polar permutation graphs. Since the foregoing example is a permutation graph, this implication is wrong.

Conversely, since the classes of edgeless graphs and P_3-free graphs are additive, monopolar graphs are additive.

POLARITY is NP-complete in general, as shown by Tyshkevich and Chernyak [91, 92]. The question becomes polynomial-time solvable, if we search for polar partitions (A, B) such that the size of every co-connected component of $G[A]$ is bounded by a constant α and the size of every connected component of $G[B]$ is bounded by a constant β [91]. This also follows from a more general result for LIST PARTITIONS given in [44]. Conversely, if we demand the bound only on β, then the problem is NP-complete [22] again. Recently, it was shown that POLARITY remains NP-complete on triangle-free graphs and on claw-free graphs [26].

In Section 5.4, we show that POLARITY is NP-complete on

- planar triangle-free graphs with maximum degree at most 3,

- planar $\{C_4, \ldots, C_g\}$-free graphs with maximum degree at most 3, for every $g \geq 4$,
- co-planar $3P_1$-free n-vertex graphs with minimum degree at least $n-4$, and
- co-planar $\{\overline{C_4}, \ldots, \overline{C_g}\}$-free n-vertex graphs with minimum degree at least $n-4$, for every $g \geq 4$.

These results have some interesting implications: On the one hand, we have NP-completeness on 3-colorable graphs, claw-free co-planar graphs, and on $\{2P_2, C_5\}$-free graphs, which includes hole-free graphs and P_5-free graphs. On the other hand, the NP-completeness on planar graphs motivates the analysis of subclasses of planar graphs that admit efficient POLARITY algorithms. In Section 4.4, we identify two classes of this kind.

The NP-completeness of MONOPOLARITY follows from a general result of Farrugia [42]. He shows that the decision problem, whether a graph admits an (A, B) vertex partition such that $G[A]$ is in the graph class Π and $G[B]$ is in the graph class Φ, is NP-complete for all additive and hereditary graph classes Π and Φ, with the single exception when both graphs have to be edgeless. Earlier, in [62], it was shown that recognizing $(1, 2)$-subcolorable cubic graphs is NP-complete on planar triangle-free graphs, which implies that MONOPOLARITY is NP-complete on planar triangle-free graphs. This also covers the NP-completeness proof for triangle-free graphs in [26]. The manuscript [23] shows that MONOPOLARITY is NP-complete on comparability graphs.

We contribute NP-completeness proofs for MONOPOLARITY on

- planar triangle-free graphs with maximum degree at most 3,
- planar $\{C_4, \ldots, C_g\}$-free graphs with maximum degree at most 3, for every $g \geq 4$,

and also on 3-colorable graphs.

UNIPOLARITY is polynomial-time solvable in general, see for example [39] and the references therein.

For POLARITY, the following positive results are known, ordered by the year of their publication; it is solvable

- in linear or polynomial time on graphs with bounded tree-width [1, 31, 33],
- in linear or polynomial time on graphs with bounded clique-width [32],
- in time $O(n)$ on cographs [38],
- in polynomial time on chordal graphs [36],
- in time $O(n)$ on line-graphs of bipartite graphs [37, 57],
- in time $O(n)$ on line-graphs [24], and
- in time $O(nm^2)$ on connected permutation graphs [35],

where n and m are the number of vertices and edges of the input graph.

We extend this list in Section 4.4 by showing that POLARITY is polynomial-time solvable on hole-free planar graphs and maximal planar graphs. In particular, the result for hole-free planar graphs complements our NP-completeness results for planar graphs and hole-free graphs.

Many of the tractability results for POLARITY, for example for chordal graphs or connected permutation graphs, analyze the polar partitions of a graph under the assumption that the graph is neither unipolar nor monopolar. Hence, the appropriate algorithms first check if the input graph is unipolar or monopolar and, if not, then check for the remaining polar partitions. This requires efficient algorithms for MONOPOLARITY. In fact, our POLARITY algorithms also work in this way. But for this purpose, we need a refined version of MONOPOLARITY that we call MONOPOLAR EXTENSION. In literature, it is also known as *monopolar multicolorings* or *list monopolar partitions*.

MONOPOLAR EXTENSION	
`Input:`	A graph $G = (V, E)$ and a precoloring (A', B').
`Question:`	Is there a monopolar partition (A, B) of G that extends (A', B')?

Clearly, MONOPOLAR EXTENSION with an empty precoloring equals MONOPOLARITY.

There are several tractability results for MONOPOLARITY and MONOPOLAR EXTENSION. The following list of results, ordered by their year of publication, contains results for both problems, indicated in parentheses. The problem is solvable

- in linear or polynomial time on graphs with bounded tree-width [1, 31, 33] (monopolar extension),
- in linear or polynomial time on graphs with bounded clique-width [32] (monopolar extension),
- in time $O(n)$ on cographs [38] (monopolarity),
- in time $O(n + m)$ on chordal graphs [36] (monopolar extension),
- in time $O(n^2m)$ on claw-free graphs [25] (monopolar extension),
- in time $O(n^4)$ on a graph class $\mathcal{G}$ that contains all hole-free graphs [27] (monopolar extension),
- in time $O(n^2m)$ on polar graphs [27] (monopolarity),
- in time $O(n)$ on line-graphs [24] (monopolarity),
- in time $O(nm)$ on permutation graphs [35] (monopolar extension), and
- in time $O(n^3)$ on claw-free graphs [26] (monopolarity),

where n and m are the number of vertices and edges of the input graph. The positive result for chordal graphs, that is, the $\{C_4, C_5, \dots\}$-free graphs, contrasts our NP-completeness result for $\{C_4, \dots, C_g\}$-free graphs for every fixed $g \geq 4$.

Our main result for MONOPOLAR EXTENSION, given in Chapter 4, is an $O(n^4)$-time algorithm that works on so called *locally A_5-$S_{2,2,2}$-defused graphs*. This graph class is quite technical, but it is a proper superclass of the class given in [27] and contains well-known graph classes like chair-free graphs and hole-free graphs and, hence, cographs, chordal graphs, claw-free graphs, line-graphs, permutation graphs, and co-comparability graphs. Notice that our result, published in [64], and the result of Churchley and Huang [27] were developed independently.

2 Efficient Domination

This chapter shows several tractability results for MINIMUM WEIGHT EFFICIENT DOMINATION. All these results concern graph classes that are characterized by forbidden induced linear forests, that is, cycle-free graphs with maximum degree at most 2, which are discussed in Section 2.1.

The given results can be classified into two categories by the technique they use. For an input graph G with vertex weights ω, the first technique seeks for every vertex v for a minimum cost efficient dominating set D_v with $v \in D_v$. This is done by partitioning the vertex set into distance levels according to v, called a hanging, and utilizing the properties of the graph class to establish an efficient search algorithm. The global minimum is found in $\min_{v \in V(G)} \{\omega(D_v)\}$. Section 2.2 presents this technique in detail and gives robust algorithms for MINIMUM WEIGHT EFFICIENT DOMINATION on $(P_2 + P_4)$-free graphs and on P_5-free graphs. The technique can also be used to design efficient algorithms for solving MINIMUM WEIGHT EFFICIENT DOMINATION on $\{P_6, S_{1,2,2}\}$-free graphs and on $\{(P_3 + P_3), S_{1,2,2}\}$-free graphs [13].

The second technique benefits from a relation between EFFICIENT DOMINATION on a graph G and MAXIMUM INDEPENDENT SET on the square of G. In Section 2.3, this relation is used to show that MINIMUM WEIGHT EFFICIENT DOMINATION is tractable on P_5-free graphs. The asymptotic runtime of this algorithm is slightly better than the runtime of the algorithm that uses the hanging technique shown in Section 2.2, but it is not robust.

The results of this chapter are published in the extended abstract [12] and the preprint [13]. Section 2.2.2 corrects some minor errors of [13].

Notions and Observations

Observation 2. *Given a graph $G = (V, E)$ with vertex weights ω, an efficient dominating set of size* 1 *and of minimum weight can be found in linear time.*

Proof. Clearly, an efficient dominating set of size 1 must be a universal vertex in G. By counting the degree of every vertex, the universal vertices

$$U := \{v \in V \mid \deg(v) = |V| - 1\}$$

can be identified in linear time. Obviously,

$$D := \{u\} \text{ with } u \in \arg\min_{u \in U} \{\omega(u)\}$$

is found in linear time. $\square$

Observation 3. *Every efficient dominating set of a graph $G = (V, E)$ that is not co-connected contains exactly one vertex.*

Proof. Let D be an efficient dominating set of G and let $\overline{H}$ and $\overline{H'}$ be two co-connected components of G. Since every vertex of $\overline{H}$ is adjacent to every vertex of $\overline{H'}$ and D is efficient, $|D \cap \overline{H}| \leq 1$. If $|D \cap \overline{H}| = 1$, the efficiency of D implies $D \cap \overline{H'} = \emptyset$. Since $\overline{H}$ and $\overline{H'}$ were chosen arbitrarily, the observation follows. $\square$

Observation 4. *Given a graph $G = (V, E)$ and a vertex subset $D \subseteq V$, it can be decided in linear time, if D is an efficient dominating set of G.*

Proof. For every vertex $d \in D$, we label every vertex in $N(d)$. This can clearly be done in time $O(|E|)$, accumulated over all $d \in D$. If we try to label a vertex twice or if we try to label a vertex of D, then D is not efficient. Finally, we check for every $v \in V \setminus D$, if v is labeled, which takes time $O(|V|)$. If this is not the case for at least one vertex, then D is not dominating. Otherwise, D clearly is an efficient dominating set of G. $\square$

Observation 5. *Let G be a graph and let v be a vertex of G. A vertex subset D is an efficient dominating set of $G - N[v]$ with $D \cap N^2(v) = \emptyset$, if and only if $D \cup \{v\}$ is an efficient dominating set of G.*

Proof. If D is an efficient dominating set of $G - N[v]$, then it dominates all vertices of G except $N[v]$. Since $D \cap N^2(v) = \emptyset$, every vertex of D has distance at least 2 to v in G. Hence, $D \cup \{v\}$ is efficient and dominating in G.

If $D' = D \cup \{v\}$ with $v \notin D$ is an efficient dominating set of G, then $D' \cap V(G - N[v]) = D$. As no vertex of $G - N[v]$ is dominated by v, D is an efficient dominating set of $G - N[v]$. $\square$

We say that an efficient dominating set D of a graph $G = (V, E)$ is *v-efficient dominating*, if $v \in D$. Consequently, an efficient dominating set D of a graph $G = (V, E)$ with vertex weights ω is *v-minimum weighted* for a vertex $v \in V$, if D has minimum weight over all v-efficient dominating sets of G.

Sometimes, we use the weight ∞ for vertices. Since we do not want to do arithmetics with infinite values, we define for a graph $G = (V, E)$ with vertex weights ω the *infinite weight* as $\infty_\omega := 1 + \sum_{v \in V} \omega(v)$. For another vertex weight function ω', we write $\omega'(D) < \infty_\omega$ for $\omega'(D) \leq \sum_{v \in V} \omega(v)$ and we write $\omega'(D) = \infty_\omega$ for $\omega'(D) > \sum_{v \in V} \omega(v)$.

Lemma 6. *If for a graph $G = (V, E)$ with vertex weights ω,* MINIMUM WEIGHT EFFICIENT DOMINATION *can be solved in time $O(s)$ for every induced subgraph $G - N[v]$, $v \in V$, then it can be solved in time $O(|V| \cdot \max\{s, |V| + |E|\})$ on G.*

Proof. If G is empty, the lemma is vacuously true. Hence, let G be non-empty.

Clearly, every minimum weighted efficient dominating set D is a v-minimum weighted efficient dominating set for all $v \in D$. Hence, to find a minimum weighted efficient dominating set of G, we compute a v-minimum weighted efficient dominating set D_v for every $v \in V$ and return one of minimum weight.

To find a v-minimum weighted efficient dominating set D, we construct

$$G_v := G - N[v] \quad \text{and} \quad \omega_v(w) := \begin{cases} \infty_\omega, & w \in N^2(v) \\ \omega(w), & \text{otherwise} \end{cases}$$

and compute a minimum weighted efficient dominating set D_v of G_v and ω_v. If it does not exist, then G_v is not efficiently dominatable and, by Observation 5, G admits no v-efficient dominating set. If $\omega_v(D_v) = \infty_\omega$, then every efficient dominating set of G_v has infinite weight, that is, contains a vertex of N^2, thus, by Observation 5, G admits no v-efficient dominating set. If $\omega_v(D_v) < \infty_\omega$, then D_v contains no vertex of N^2 and, by Observation 5, $D := D_v \cup \{v\}$ is a v-efficient dominating set of G. Since ω and ω_v only differ in vertices of N^2, we have $\omega(D) = \omega_v(D_v)$.

Assume that G admits a v-efficient dominating set D' with $\omega(D') < \omega(D)$. By Observation 5, there is an efficient dominating set D'_v of G_v with $N^2 \cap D'_v = \emptyset$. Since ω and ω_v only differ in vertices of N^2, we have $\omega(D') = \omega_v(D'_v)$, which implies $\omega_v(D'_v) < \omega_v(D_v)$—this is a contradiction because D_v is a minimum weighted efficient dominating set of G_v. Hence, D is a v-minimum weighted efficient dominating set of G.

Since G_v and ω_v can be constructed in time $O(|V| + |E|)$, solving MINIMUM WEIGHT EFFICIENT DOMINATION on G_v and ω_v takes time $O(s)$, and we do this for every vertex $v \in V$, the stated runtime of $O(|V| \cdot \max\{s, |V| + |E|\})$ follows. $\square$

For a graph class $\mathcal{C}$, a graph $G = (V, E)$ is *nearly* $\mathcal{C}$, if for every $v \in V$ the graph $G - N[v]$ is in $\mathcal{C}$. For nearly $\mathcal{C}$ graphs, Lemma 6 implies:

Corollary 7. *If* MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable on every graph $G = (V, E)$ of class $\mathcal{C}$ in time $O(s(|V|, |E|))$, then it is solvable on every nearly $\mathcal{C}$ graph $G' = (V', E')$ in time $O(|V'| \cdot \max\{s(|V'|, |E'|), |V'| + |E'|\})$.*

A proper vertex subset $H \subset V$ of a graph $G = (V, E)$ with $2 \leq |H|$ is called *homogeneous*, if every vertex of $V \setminus H$ is either adjacent to all vertices of H or adjacent to no vertex of H. A graph is called *prime*, if it contains no homogeneous set. A homogeneous set H is *maximal*, if no other homogeneous set properly contains H. It is well known that in a connected and co-connected graph G, the maximal homogeneous sets are pairwise disjoint and can be determined in linear time (see for example [75]).

Observation 8. *Let $G = (V, E)$ be a connected graph with vertex weights ω, let H be a maximal homogeneous set of G and let $U \subseteq H$ contain all H-universal vertices of H. If D is a minimum weighted efficient dominating set of G, then $|D \cap H| \leq 1$ and $D \cap H \subseteq \arg\min\{\omega(u) \mid u \in U\}$.*

Proof. Let D be a minimum weighted efficient dominating set of G. Assume that there are vertices h and h' with $\{h, h'\} \subseteq H \cap D$. Since $H \neq V$, H is homogeneous, and G is connected, there is at least one vertex $y \in V \setminus H$ that is adjacent to every vertex of H, in particular to h and h'—this is a contradiction to the efficiency of D. Hence, $|D \cap H| \leq 1$.

Assume that $D \cap H = \{h\}$. First, assume that $h \notin U$. There is a vertex $h' \in H$ with $h \cdots h'$. Since D is dominating, there is a vertex $y \in D$ with $h' - y$. Since $|D \cap H| \leq 1$, we have $y \in V \setminus H$. This implies $h - y$ because H is homogeneous—this is a contradiction to the efficiency of D. Hence, $h \in U$. Now assume that there is $h' \in U$ with $\omega(h') < \omega(h)$. Since $N[h] = N[h']$ for all $\{h, h'\} \in U$, clearly $D' = D \setminus \{h\} \cup \{h'\}$ is an efficient dominating set of G with $\omega(D') < \omega(D)$—this is a contradiction to the minimum size of D.

Since all vertices of H can be dominated by a vertex of $V \setminus H$, $H \cap D = \emptyset$ is also possible. $\square$

Definition 1. *Let $G = (V, E)$ be a connected and co-connected graph with vertex weights ω and let $\mathcal{H} = \{H_1, H_2, \ldots, H_k\}$ be the maximal homogeneous* *sets of G. Let $\mathcal{U} \subseteq \mathcal{H}$ contain every homogeneous set H of G that contains at least one H-universal vertex. For every $H_i \in \mathcal{U}$, let $h_i \in V(H_i)$ be a H_i-universal vertex of minimum weight. For every $H_i \in \mathcal{H} \setminus \mathcal{U}$, let $h_i \in V(H_i)$ be an arbitrarily chosen vertex of H_i. The* packed *graph is defined as*

$$G^p := G[(V \setminus (H_1 \cup H_2 \cup \cdots \cup H_k)) \cup \{h_1, h_2, \ldots, h_k\}]$$

$$\omega^p(v) := \begin{cases} \infty_\omega, & \text{if } v = h_i \text{ and } H_i \notin \mathcal{U} \\ \omega(v), & \text{otherwise} \end{cases}.$$

Since all homogeneous sets are contracted to a single vertex, G^p clearly is prime.

Lemma 9. *Let $G = (V, E)$ be connected and co-connected graph with vertex weights ω. There is a minimum weighted efficient dominating set of G of size $\Omega < \infty_\omega$, if and only if there is a minimum weighted efficient dominating set of G^p of size Ω.*

Proof. Let D be a minimum weighted efficient dominating set of G. Clearly, we have $\omega(D) < \infty_\omega$ because $D \subseteq V$. Let $H_1, \ldots, H_k$ be the maximal homogeneous sets of G. Let D^p contain

- $D \setminus (H_1 \cup H_2 \cup \cdots \cup H_k)$ and
- $H_i \cap V(G^p)$, if $H_i \cap D \neq \emptyset$, for all $i \in \{1, \ldots, k\}$.

Clearly, we have $D^p \subseteq V(G^p)$. Since D^p contains a vertex of a maximal homogeneous set H of G, only if D contains a vertex of H, by Observation 8 we know that D^p contains no vertex v with $\omega^p(v) = \infty_\omega$ and, hence, $\omega^p(D^p) < \infty_\omega$. Also by Observation 8, we know that D contains at most one vertex h of every homogeneous set H of G. Furthermore, h is H-universal and of minimum weight over all H-universal vertices. In D^p, h is substituted by a vertex that has the same neighborhood in G and that has the same weight as h. Hence, and since G^p is an induced subgraph of G, D^p is an efficient dominating set of G^p with $\omega^p(D^p) = \omega(D)$.

Let D^p be a minimum weighted efficient dominating set of G^p with $\omega^p(D^p) < \infty_\omega$. Since $\omega^p(D^p) < \infty_\omega$, we know that D^p contains no vertex of weight ∞_ω and, hence, for every $d^p \in D^p$ we have $\omega^p(d^p) = \omega(d^p)$. The set D^p is efficient and dominating in G because every vertex of $V(G) \setminus V(G^p)$ has the same neighborhood as some vertex of $V(G^p)$ and assuming that D^p is not efficient or not dominating in G yields that D^p is not efficient or not dominating in G^p. $\square$

Since finding all maximal homogeneous sets and constructing G^p can be done in linear time, from Lemma 9 follows:

Corollary 10. *Let $G = (V, E)$ be a connected and co-connected graph. If* MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in time $O(s)$ on G^p, then it can be solved in time $O(\max\{s, |V| + |E|\})$ on G.*

2.1 Graphs Without Induced Linear Forests

EFFICIENT DOMINATION is known to be NP-complete on several graph classes, including bipartite graphs and chordal graphs [94]. Since bipartite graphs are a subset of the triangle-free graphs, we have:

Corollary 11 (of Theorem 1 in [94])**.** EFFICIENT DOMINATION *is NP-complete on triangle-free graphs.*

Theorem 1 (Theorem 2 in [94])**.** EFFICIENT DOMINATION *is NP-complete on chordal graphs.*

Another result originates from the NP-completeness of EFFICIENT EDGE DOMINATION [49]: By definition, a graph G admits an efficient edge dominating set, if and only if its line graph $L(G)$ admits an efficient dominating set. Hence, EFFICIENT DOMINATION is NP-complete on the class of line graphs. Since the line graphs are claw-free, we have:

Corollary 12 (by [5])**.** EFFICIENT DOMINATION *is NP-complete on claw-free graphs.*

Recall that linear forests are the graphs that do not contain a cycle or a claw. Corollaries 11 and 12 and Theorem 1 together imply:

Corollary 13. *Let F be a graph. If F is not a linear forest, then* EFFICIENT DOMINATION *is NP-complete on F-free graphs.*

This gives rise to the question how EFFICIENT DOMINATION behaves on graph classes that are defined by a single forbidden induced linear forest.

We can find a first intractability result by having a closer look at [94]. The reduction given therein constructs a graph consisting of independent vertex sets X, S, and A of equal size such that $S \cup A$ induces a 1-regular graph, and an EXACT 3-COVER instance is encoded in the edges between X and S. The reduction also works if X is a clique, which is used for showing NP-completeness on chordal graphs. In that case, the graph is also $(P_3 + P_3)$-free and, hence, we have:

Theorem 2 (similar to Theorem 2 in [94]). Efficient domination *is NP-complete on* $(P_3 + P_3)$*-free graphs.*

In fact, this is the only NP-completeness result that we know for F-free graphs, if F is a linear forest.

Conversely, we collect some tractability results that are either trivial or follow from known results. A useful tool for this is the following lemma:

Lemma 14. *Let* F *be a linear forest. If* minimum weight efficient domination *is polynomial-time solvable on* F*-free graphs, then it is also polynomial-time solvable on* $(F + P_1)$*-free graphs. Conversely, if* efficient domination *is NP-complete on* F*-free graphs, then it also is NP-complete on* $(F + P_1)$*-free graphs.*

Proof. By definition, every $(F + P_1)$-free graph is nearly F-free. Hence, Corollary 7 implies the first statement of the corollary. Since F-free graphs are a subclass of $(F + P_1)$-free graphs, the second statement holds, too. □

Clearly, minimum weight efficient domination is linear-time solvable on $(P_1 + P_1)$-free graphs and P_2-free graphs, that is, on complete and on edgeless graphs. Although every edgeless or complete graph is efficiently dominatable, notice that we need linear time to calculate the weight of the efficient dominating set. Hence, for every F-free graph where F has at most two vertices, minimum weight efficient domination is linear-time solvable.

Using Lemma 14, we have that minimum weight efficient domination is polynomial-time solvable on $3P_1$-free graphs and $(P_2 + P_1)$-free graphs. Actually, using the following simple observations, we can easily derive linear-time algorithms for these classes:

Observation 15. *If* $G = (V, E)$ *is an efficiently dominatable* $3P_1$*-free graph without a universal vertex, then* G *is co-bipartite. Let* $V = X \cup Y$ *be a partition of* G *into cliques. The vertex subset* $D \subseteq V$ *is an efficient dominating set of* G*, if and only if* $D = \{x, y\}$ *for* $x \in X$ *and* $y \in Y$ *with* $E(x, Y) = \emptyset$ *and* $E(y, X) = \emptyset$*.*

Proof. Since G is $3P_1$-free, every independent set of G has at most 2 vertices and, hence, every efficient dominating set D has at most 2 vertices. If G has no universal vertex, we have $|D| = 2$, say $D = \{x, y\}$. Since D is efficient, $N(x) \cap N(y) = \emptyset$. This implies $V = N[x] \cup N[y]$ because if there is a vertex $v \in V \setminus (N[x] \cup N[y])$, then $\{v, x, y\}$ induces a $3P_1$ in G—this is a contradiction. Furthermore, $N[x]$ and $N[y]$ are cliques because if there are two non-adjacent vertices in one of the sets, say $\{x', x''\} \in N(v)$, then $\{y, x', x''\}$ induces a $3P_1$ in G—this is a contradiction. Hence, G is co-bipartite.

The fact that every choice of two vertices, $x \in X$ and $y \in Y$, which fulfill $E(x, Y) = \emptyset$ and $E(y, X) = \emptyset$ forms an efficient dominating set of G follows from the previous considerations. □

Recognizing co-bipartite graphs can be done in linear time because, if a graph $G = (V, E)$ is co-bipartite, then $|E| \geq |V|^2$ and, hence, we can compute $\overline{E}$ in linear time and test if $\overline{G}$ is bipartite. Clearly, for a bipartition of $\overline{G} = (X \cup Y, \overline{E})$, finding all X-universal vertices of Y and all Y-universal vertices of X and choosing a pair of minimum weight can be done in linear time, too. Hence, we can solve MINIMUM WEIGHT EFFICIENT DOMINATION in linear time on $3P_1$-free graphs.

Observation 16. *If $G = (V, E)$ is an efficiently dominatable $(P_2 + P_1)$-free graph, then G is either edgeless or contains a universal vertex.*

Proof. If G is edgeless, clearly, V is an efficient dominating set. Otherwise, assume that G has an efficient dominating set $D \subset V$ with at least two vertices, say $\{x, y\} \subseteq D$. Without loss of generality assume that x has a neighbor x'. Since D is efficient, we have $x' \cdots y$ and, hence, $\{x, x', y\}$ induces a $(P_2 + P_1)$ in G—this is a contradiction. □

Testing a graph for being edgeless is easy. Finding a universal vertex of minimum weight can be done by counting the degree of every vertex and compare it to $|V| - 1$ in linear time. Hence, MINIMUM WEIGHT EFFICIENT DOMINATION is solvable in linear time on $(P_2 + P_1)$-free graphs.

The class of P_3-free graphs contains exactly the disjoint unions of complete graphs. Since efficiently dominatable graphs are additive, we can solve the problem in linear-time on P_3-graphs by solving it on the connected components. This proves:

Theorem 3. *On F-free graphs, where F is a linear forest on at most three vertices,* MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in linear time.*

When considering linear forests on four vertices, the cases that are not covered by Lemma 14 and Theorem 3 are $(P_2 + P_2)$-free graphs and P_4-free graphs. For $(P_2 + P_2)$-free graphs and even for mP_2-free graphs for every fixed m it is known that the number of inclusion-maximal independent sets is polynomial in the number of vertices [3, 41, 85]. Since the set of all inclusion-maximal independent sets of a graph $G = (V, E)$ can be found in time $O(|V| \cdot |E| \cdot \mu)$ [90], where μ is the number of all inclusion-maximal independent sets, MINIMUM WEIGHT EFFICIENT DOMINATION can be solved in polynomial time by testing every inclusion-maximal independent set for being an efficient dominating set by Observation 4 and choosing the one of minimum weight. For the class of P_4-free graphs, better known as cographs, it is known that every graph is either the P_1 or not connected or the union of two cographs H and H' extended by all possible edges between $V(H)$ and $V(H')$. If G is not connected, then MINIMUM WEIGHT EFFICIENT DOMINATION can be solved on the connected components independently. If G is not co-connected, then every pair of vertices has distance at most 2 and, hence, every efficient dominating set contains exactly one vertex. Thus, by Observation 2, the problem can be solved in linear time on cographs. This proves:

Theorem 4. *On F-free graphs, where F is a linear forest on at most four vertices,* MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in polynomial time.*

If F is a linear forest on five vertices, Lemma 14 and Theorem 4 give tractability results for all classes except P_5-free graphs and $(P_3 + P_2)$-free graphs. In both cases we are not aware of known results that imply NP-completeness or tractability of EFFICIENT DOMINATION. Therefore, Sections 2.2.1 and 2.3.1 give two polynomial-time algorithms for P_5-free graphs and Section 2.2.2 gives a polynomial-time algorithm for $(P_4 + P_2)$-free graphs, hence, also for $(P_3 + P_2)$-free graphs. With these results, we have:

Theorem 5. *On F-free graphs, where F is a linear forest on at most five vertices,* MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in polynomial time.*

Considering forbidden induced linear forests on six vertices, there are four cases that are not covered by Lemma 14 and Theorem 5. In the case of $(P_3 + P_3)$-free graphs, we already know NP-completeness (Theorem 2). The second case, the $3P_2$-free graphs, admit a polynomial-time algorithm due to the bounded number of maximal independent sets as discussed earlier for $(P_2 + P_2)$-free graphs. The third case are the $(P_4 + P_2)$-free graphs. As already mentioned, Section 2.2.2 shows that MINIMUM WEIGHT EFFICIENT DOMINATION is tractable on this class. The fourth case, that is, the P_6-free graphs, is the only one where the complexity of EFFICIENT DOMINATION remains open.

Theorem 6. *On F-free graphs, where F is a linear forest on at most six vertices, except $F = (P_3 + P_3)$ and $F = P_6$,* MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in polynomial time.*

Since EFFICIENT DOMINATION is NP-complete on $(P_3 + P_3)$-free graphs, it is clearly also NP-complete on P_k-free graphs for every $k \geq 7$ and on $(P_k + P_\ell + F')$-free graphs for every $k \geq 3$, every $\ell \geq 3$, and every linear forest F'. Hence, besides the P_6-free case, the complexity of EFFICIENT DOMINATION on F-free graphs is still unknown for $(P_5 + P_2)$-free graphs, $(P_6 + P_2)$-free graphs, and for $(P_k + mP_2)$-free graphs for every $k \geq 3$ and every $m \geq 2$. If these questions can be solved, the classification of the complexity of EFFICIENT DOMINATION with respect to graphs without induced linear forests would be complete.

2.2 Hangings

For a graph $G = (V, E)$ and a vertex $v \in V$, we call $N^1(v), N^2(v), \ldots$ the *distance levels* of v. Clearly, there is a $k < |V|$ such that

$$\{v\} \cup N^1(v) \cup N^2(v) \cup \cdots \cup N^k(v)$$

is a partition of V. We call this partition the *hanging* of G for v and denote it by

$$\mathcal{N}_v := \left\{ N^1(v), N^2(v), \ldots, N^k(v) \right\}.$$

The algorithms shown in this section solve MINIMUM WEIGHT EFFICIENT DOMINATION in the following way: Given a graph $G = (V, E)$, check if it is connected and co-connected. If not, either solve the problem on the connected components or find and return the trivial solution containing a single vertex. Otherwise, continue on the packed graph, which is justified by Lemma 9. Determine the hanging for every vertex $v \in V$, find a v-minimum weighted efficient dominating set D_v, and return the best of all these efficient dominating sets. Since every efficient dominating set of a connected non-empty graph contains at least one vertex, this procedure is justified by Lemma 6. Table 2.1 shows the appropriate algorithm. It contains the call of a function `Robust-`$\mathcal{C}$`-`v`-MWED`, which is defined as follows:

Definition 2. *The procedure* `Robust-`$\mathcal{C}$`-`v`-MWED` *takes a prime graph $G = (V, E)$, a vertex $v \in V$ and the hanging on this vertex, assuming that $N^3(v) \neq \emptyset$, as arguments and either returns a set $D_v \subseteq V$ or states that $G \notin \mathcal{C}$. If G*

is v-efficiently dominatable, then a set D_v is returned which is a v-minimum weighted efficient dominating set of G. If G is not v-efficiently dominatable, then a set D_v is returned which is no efficient dominating set at all.

Since the returned set D_v is efficient dominating in G, if and only if it is a v-minimum weighted efficient dominating set of G, Step 6 of the framework has to check for every returned set whether it is an efficient dominating set and discards invalid candidates. This check is no performance drawback for the algorithm because deciding if a vertex subset is efficient dominating can be done in linear time by Observation 4.

With the considerations from above, Observations 2 and 3 and Corollary 10, one can easily check:

Lemma 17. *If* `Robust-`$\mathcal{C}$`-`v`-MWED` *fulfills Definition 2 and can be computed in time $O(s)$ for a graph $G = (V, E) \in \mathcal{C}$, then* `Robust-`$\mathcal{C}$`-MWED` *is correct and runs in time $O(|V| \cdot \max\{s, |V| + |E|\})$.*

As already mentioned, the following sections give implementations of `Robust-`$\mathcal{C}$`-`v`-MWED` for $\mathcal{C}$ fixed to P_5-free graphs and $(P_4 + P_2)$-free graphs. Therefore, the hanging for a vertex v is analyzed under the assumption that G is in the graph class $\mathcal{C}$ and that G admits a v-efficient dominating set.

Hence, let $G = (V, E)$ be a prime graph with vertex weights ω, let $v \in V$ be a vertex of G, let $D \subseteq V$ with $v \in D$, and let $N^3(v) \neq \emptyset$. For brevity, we write N^i instead of $N^i(v)$.

The following two sections are structured as follows: We collect properties for G and D and show in subsequent lemmas under which conditions the properties are fulfilled. Then, we use these insights to implement `Robust-`$\mathcal{C}$`-`v`-MWED` efficiently for the respective graph class $\mathcal{C}$.

A trivial property following directly from the definition, that is independent from $\mathcal{C}$ and true, whenever G admits a v-efficient dominating set, is:

Property 1. *$D \cap N^1 = D \cap N^2 = \emptyset$.*

This property is obvious and, therefore, never explicitly mentioned when used.

2.2.1 P_5-free Graphs

Property 2. *We have $V = \{v\} \cup N^1 \cup N^2 \cup N^3$, that is, $N^4 = N^5 = \cdots = \emptyset$.*

Property 3.

Algorithm: `Robust-`$\mathcal{C}$`-MWED`
Input: A prime graph $G = (V, E)$ with vertex weights ω.
Output: A minimum weighted efficient dominating set of G or the statement that G admits no efficient dominating set or the statement that $G \notin \mathcal{C}$.

1. If G is not connected and $G_1, G_2, \ldots, G_k$ are the connected components of G:
    1. Compute $D_i :=$ `Robust-`$\mathcal{C}$`-MWED`(G_i) for all $i \in \{1, \ldots, k\}$.
    2. If one of these computation stops, `STOP` for the same reason.
    3. `Return` $D_1 \cup D_2 \cup \cdots \cup D_k$.
2. If G is not co-connected, calculate the set U of universal vertices of G and `Return` $\{u\}$ for some $u \in \arg\min \{\omega(u) \mid u \in U\}$.

— *G is connected and co-connected.* —

3. Construct G^p and set $G = (V, E) := G^p$.
4. Set $\mathcal{D} := \emptyset$.
5. For every vertex $v \in V$:
    1. Determine the hanging $\mathcal{N}_v$.

    — *Since G is co-connected, $N^2(v) \neq \emptyset$.* —

    2. If $N^3(v) = \emptyset$, set $D_v := \emptyset$.

        /* *We need vertices in $N^3(v)$ to dominate the vertices of $N^2(v)$.* */

    3. Else, set $D_v :=$ `Robust-`$\mathcal{C}$`-`v`-MWED`$(G, v, \mathcal{N}_v)$.
    4. Set $\mathcal{D} := \mathcal{D} \cup \{D_v\}$.
6. For every $D \in \mathcal{D}$ check if D is an efficient dominating set and, if so, calculate its weight. If not, remove D from $\mathcal{D}$.
7. If $\mathcal{D}$ contains no efficient dominating set at all, `STOP:` G `admits no efficient dominating set`.
8. `Return` D for some $D \in \arg\min_{D \in \mathcal{D}} \{\omega(D)\}$.

Table 2.1: A framework for solving MINIMUM WEIGHT EFFICIENT DOMINATION using hangings.

(i) For every two vertices $\{y, z\} \subseteq N^3$ *with* $y{-}z$*, we have* $N(y) \cap N^2 = N(z) \cap N^2$*, that is,* y *and* z *have the same neighborhood in* N^2*.*

For every connected component H *of* $G[N^3]$*:*

(ii) for every two vertices $\{y, z\} \subseteq V(H)$*, we have* $N(y) \cap N^2 = N(z) \cap N^2$*, that is,* y *and* z *have the same neighborhood in* N^2 *and*

(iii) H *contains at most one* H*-universal vertex.*

Lemma 18. *If* G *is* P_5*-free, then it fulfills Properties 2 and 3.*

Proof. Property 2 directly follows from the P_5-freeness.

For Property 3.i, let $\{y, z\} \subseteq N^3$ with $y{-}z$. Assume without loss of generality that there is a vertex $x \in N^2$ with $x{-}y$ and $x \cdots z$. Let w be a neighbor of x in N^1. Then $\{v, w, x, y, z\}$ induces a P_5 in G—this is a contradiction. Property 3.ii follows by applying this argumentation on every edge of H.

For Property 3.iii, assume that there are two different H-universal vertices in $V(H)$, say u and u'. Both, u and u', have the same neighborhood in N^3 because they are adjacent to all vertices of H and to no other vertex of N^3. As shown above, they also have the same neighborhood in N^2. By definition, they have no other neighbors in G, in particular there is at least one common non-neighbor, for example v. This means that $\{u, u'\}$ is a homogeneous set of G—this is a contradiction to its primality. $\square$

Property 4. *Every connected component* H *of* $G[N^3]$ *contains one* H*-universal vertex* u *and* $D \cap H = \{u\}$*.*

Lemma 19. *If* G *admits a* v*-efficient dominating set* D *of* G *and* G *fulfills Properties 2 and 3, then* G *and* D *fulfill Property 4.*

Proof. Since $N^1 \cap D = N^2 \cap D = N^4 = N^5 = \cdots = \emptyset$, all vertices of N^3 must be dominated by vertices of N^3. Let $y \in N^3$ and let H be the connected component of $G[N^3]$ that contains y. Clearly, y has at least one neighbor in N^2, say x. Hence, by Property 3.ii, every vertex $y' \in V(H)$ is adjacent to x. This means that D contains exactly one vertex of H because if $V(H) \cap D = \emptyset$, then D is not dominating, and if $|V(H) \cap D| > 1$, then D is not efficient. Since D is dominating, the vertex u in $V(H) \cap D$ is H-universal. By Property 3.iii, there is at most one H-universal vertex in $V(H)$ and the lemma follows. $\square$

Property 5. *For every vertex* $x \in N^2$*, there is a connected component* H *of* $G[N^3]$ *with* $N(x) \cap N^3 = V(H)$*.*

Lemma 20. *If G admits a v-efficient dominating set D of G and G and D fulfill Properties 3 and 4, then G fulfills Property 5.*

Proof. That there is a connected component H of $G[N^3]$ such that $V(H) \subseteq (N(x) \cap N^3)$ follows from Property 3.ii. Assume that x has also a neighbor in a different connected component H' of $G[N^3]$. Again, by Property 3.ii, $V(H') \subseteq (N(x) \cap N^3)$. Hence, by Property 4, x has two neighbors in D—this is a contradiction to the efficiency of D. □

With Lemmas 18 to 20, we know that, if G is P_5-free and admits a v-efficient dominating set D, then G and D fulfill Properties 2 to 5. To develop an algorithm, we also need the converse:

Lemma 21. *If G and D fulfill Properties 2 to 5, then D is a v-minimum weighted efficient dominating set of G.*

Proof. Clearly, v has distance 3 to every other vertex of D and dominates all vertices of $N^1[v]$. Since D contains an H-universal vertex of every connected component H of $G[N^3]$, all vertices in N^3 are dominated.

Assume that D is not efficient. That means, that there are two different vertices $\{u, u'\} \subseteq D$ with distance 1 or 2. We already know that $u \neq v$ and $u' \neq v$. Hence, by Property 4, u is the H-universal vertex of a connected component H of $G[N^3]$ and u' is the H'-universal vertex of a different connected component H' of $G[N^3]$. This implies that $\mathrm{dist}(u, u') = 2$ and that u and u' have a common neighbor $y \in N^2$—this is a contradiction to Property 5.

Assume that D is not dominating. Then there is a vertex $x \in N^2$ without a neighbor in D. By Property 5, the neighborhood of x in N^3 induces a connected component of $G[N^3]$ and, by Property 4, every connected component of $G[N^3]$ contains a vertex of D—this is a contradiction.

Since D is uniquely defined, it is of minimum weight over all v-efficient dominating sets of G. □

The previous lemmas straightforwardly describe how to find for a given graph $G = (V, E)$ and a vertex $v \in V$ a v-minimum weighted efficient dominating set. Table 2.2 shows the corresponding algorithm.

Lemma 22. *The procedure* `Robust-`P_5`-free-`v`-MWED` *is correct in the sense of Definition 2 and runs in time $O(|V| + |E|)$.*

Procedure: `Robust-`P_5`-free-`v`-MWED(`$G = (V, E)$`, `v`, `$\mathcal{N}_v$`)`
Input: As defined in Definition 2.
Output: As defined in Definition 2.

1. If $N^4 \neq \emptyset$, then `STOP:` G `is not` P_5`-free`.
2. Determine the connected components $H_1, H_2, \ldots, H_k$ of $G[N^3]$.
3. Check if G fulfills Property 3. If not, then `STOP:` G `is not` P_5`-free.`
4. Check if G can fulfill Property 4. If not, then `Return` $\emptyset$.

   / G admits no v-efficient dominating set. */*

5. Let u_i be the H_i-universal vertex of H_i for every $i \in \{1, \ldots, k\}$.
6. `Return` $\{v, u_1, \ldots, u_k\}$.

   / This is efficient and dominating, if and only if Property 5 holds. */*

Table 2.2: The procedure `Robust-`P_5`-free-`v`-MWED`

Proof. The correctness of Steps 1, 3, and 4 follows from Lemmas 18 and 19. Just before Step 6, G and $D = \{v, u_1, \ldots, u_k\}$ clearly fulfill Properties 2 to 4. If G violates Property 5, then, by Lemma 20, G admits no v-efficient dominating set and, hence, D is not an efficient dominating set of G. Conversely, if G fulfills Property 5, then, by Lemma 21, D is a v-efficient dominating set of G. Since D is the unique set that fulfills Property 4, it is a v-minimum weighted efficient dominating set of G.

The connected components of $G[N^3]$ can be computed in time $O(|V|+|E|)$ using depth-first-search or breadth-first-search. We can label every vertex of N^3 with the connected component it belongs to.

Property 3 can be checked in the following way: For every vertex $y \in N^3$, label every vertex of $N(y) \cap N^2$ with H, the connected component of $G[N^3]$ that y belongs to. This takes at most $O(|E|)$ time because we can consider every edge xy of G and check if $x \in N^2$ and $y \in N^3$ and, if so, label x with the connected component y is labeled with. After that, count the number of labels of the same kind for every vertex $x \in N^2$ and compare this number to the size of the connected component of $G[N^3]$ that the label corresponds to. If the numbers are not equal, G clearly violates Property 3. This takes at most $O(|E|)$ time because the sum of all labels assigned to all vertices of N^2 is bounded by the number of edges of G.

The graph G can fulfill Property 4, if every connected component H of $G[N^3]$ contains exactly one H-universal vertex. This can be checked in time $O(|E|)$ by initializing a degree counter for every vertex of N^3 with zero, considering every edge of G, checking if both endpoints of the edge are in N^3 and, if so, increasing the degree counter for both endpoints by one, and, finally, checking for every connected component H of $G[N^3]$, if there is exactly one vertex in H with degree $|V(H)|$. This way, we already identified the desired H-universal vertex for every connected component H of $G[N^3]$. $\square$

Together with Lemma 17, Lemma 22 implies:

Theorem 7. MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in time $O(nm)$ on P_5-free graphs in a robust way, where n is the number of vertices and m is the number of edges of the input graph.*

2.2.2 $(P_4 + P_2)$-free Graphs

Before giving the main result of this section, namely a robust $O(nm)$-time algorithm for MINIMUM WEIGHT EFFICIENT DOMINATION on $(P_4 + P_2)$-free graphs that uses the mentioned hanging framework, we show a much simpler algorithm that solves the problem in time $O(n^2m)$.

A Simple Algorithm

The basic idea is that, unless the input graph G contains a universal vertex, every efficient dominating set D has at least two vertices that are the endpoints of an induced P_4. Assume that $x{-}x'{-}y'{-}y$ is induced in G. The absence of an induced $(P_4 + P_2)$ implies that $N^3(\{x, y\})$ is independent and $N^4(\{x, y\})$ is empty. If $\{x, y\} \subseteq D$, the only way to dominate the vertices of $N^2(\{x, y\})$ and $N^3(\{x, y\})$ is

$$D = \{x, y\} \cup N^3(\{x, y\}).$$

This immediately yields an $O(n^2m)$-time algorithm that checks every pair x, y of vertices for being the endpoints of an induced P_4 and, if so, tests if $\{x, y\} \cup N^3(\{x, y\})$ is an efficient dominating set, which can be done in linear time by Observation 4. After that, it returns an efficient dominating set of minimum weight over all solutions.

A Faster Algorithm

Let $R = N^3 \cup N^4 \cup N^5$. Let $H_1, \dots, H_k$ be the connected components of $G[R]$ and, for every H_i, let $U_i \subseteq V(H_i)$ be the set of H_i-universal vertices of H_i.

Notice that $U_i \subseteq (N^3 \cup N^4)$ because, if a connected component H_i of $G[R]$ has vertices in N^5, then, since $E(N^3, N^5) = \emptyset$, all H_i-universal vertices are in N^4. Let $\mathcal{H}^3 \cup \mathcal{H}^{3,4} \cup \mathcal{H}^4$ be a partition of $\{H_1, \ldots, H_k\}$, where

- $H_i \in \mathcal{H}^3$, if $U_i \neq \emptyset$ and $U_i \subseteq N^3$, and
- $H_i \in \mathcal{H}^4$, if $U_i \neq \emptyset$ and $U_i \subseteq N^4$, and
- $H_i \in \mathcal{H}^{3,4}$, if $U_i = \emptyset$ or if $U_i \cap N^3 \neq \emptyset$ and $U_i \cap N^4 \neq \emptyset$.

For this partition, assume, without loss of generality, that, for every pair H_i, H_j of connected components of $G[R]$,

- $H_i \in \mathcal{H}^3$ and $H_j \in (\mathcal{H}^{3,4} \cup \mathcal{H}^4)$ implies $i < j$ and
- $H_i \in \mathcal{H}^{3,4}$ and $H_j \in \mathcal{H}^4$ implies $i < j$.

Often, we refer to the set $(\mathcal{H}^3 \cup \mathcal{H}^{3,4})$, hence, we define

$$\mathcal{H} := \left(\mathcal{H}^3 \cup \mathcal{H}^{3,4}\right),$$

that is, $\mathcal{H}$ contains every connected component H_i of $G[R]$ such that U_i is either empty or contains a vertex of N^3.

A vertex $x \in N^2$ is called *U_i-dependent*, if $N(x) \cap U_j = \emptyset$ for all $i \neq j$. We say that a vertex $x \in N^2$ is *U_i-sparse*, if there are at least two vertices in U_i that are not adjacent to x. Notice that, if $G[R]$ is connected, then all vertices of N^2 are U_1-dependent. Also, if a vertex $x \in N^2$ has no neighbor in any U_i, then x is U_i-dependent for every $i \in \{1, \ldots, k\}$.

For every $H_i \in (\mathcal{H} \cup \mathcal{H}^4)$, let $D_i \subseteq N^2$ contain all U_i-dependent vertices and let

$$U_i' = U_i \cap \bigcap_{x \in D_i} N(x),$$

that is, if there are no U_i-dependent vertices, then $U_i' = U_i$, and otherwise, U_i' contains only the vertices of U_i that are adjacent to all U_i-dependent vertices. Notice that, if there is a vertex $x \in N^2$ with no neighbor in any U_i, then $U_i' = \emptyset$ for every $i \in \{1, \ldots, k\}$. If there is no such vertex, then notice that, for every $H_i \in \mathcal{H}^4$, we have $D_i = \emptyset$ and, hence, $U_i' = U_i$.

Property 6. *(i) $N^6 = N^7 = \cdots = \emptyset$ and $G[R]$ is a cograph. (ii) If a vertex $x \in N^2$ is U_i-sparse, then it is U_i-dependent.*

Lemma 23. *If G is $(P_4 + P_2)$-free, then G fulfills Property 6.*

Proof. The first statement immediately follows from the $(P_4 + P_2)$-freeness.

For the second statement, let $x \in N^2$ be a U_i-sparse vertex. There are vertices $\{y, y'\} \subseteq U_i$ with $x \cdots y$ and $x \cdots y'$. Assume that x is not U_i-dependent, that is, x has a neighbor $y'' \in U_j$ for some $j \neq i$. Let w be a common neighbor of v and x in N^1. Since y'' lies in a different connected component of $G[R]$ than y and y'', clearly $\{vwxyy'y''\}$ induces a $(P_4 + P_2)$ in G—this is a contradiction. □

Property 7. *For every $i \in \{1, \dots, k\}$, we have $U'_i \neq \emptyset$. Furthermore, $\mathcal{H} \neq \emptyset$.*

Property 8. *For every $i \in \{1, \dots, k\}$, we have $V(H_i) \cap D = \{u_i\}$ for some $u_i \in U'_i$.*

Lemma 24. *If G admits a v-efficient dominating set D and G fulfills Property 6, then G fulfills Property 7 and D fulfills Property 8.*

Proof. Let $H_i \in (\mathcal{H} \cup \mathcal{H}^4)$ be chosen arbitrarily. By Property 6.i, H_i is a connected cograph. Since cographs are P_4-free, every efficient dominating set of H_i is of size 1. Hence, there must be an H_i-universal vertex u_i to dominate all vertices of H_i because all other vertices of G that are adjacent to vertices of H_i are in N^2, which cannot contain vertices of D. This means, that $u_i \in U_i \cap D$. Assume that $u_i \in (U_i \setminus U'_i)$, that is, there is a U_i-dependent vertex $x \in N^2$ with $x \cdots u_i$. Since $u_i \in D$, x is not dominated because all vertices that can dominate x are in U_i—this is a contradiction, since D is dominating and efficient. Hence, it must be $u_i \in U'_i$.

Assume that $\mathcal{H} = \emptyset$, that is, all vertices of $D \cap R$ are in N^4. Since $D \cap N^2 = \emptyset$, the vertices of N^2 are not dominated by D—this is a contradiction, hence, we have $\mathcal{H} \neq \emptyset$. □

Property 9. *For every $H_i \in \mathcal{H}^4$, we have $|V(H_i) \setminus N^3| = 1$, hence, $|V(H_i) \cap N^4| = |U'_i| = 1$ and $V(H_i) \cap N^5 = \emptyset$.*

Lemma 25. *If G is $(P_4 + P_2)$-free and G fulfills Property 7, then G fulfills Property 9.*

Proof. Assume that there is a connected component $H_i \in \mathcal{H}^4$ with $\{z, z'\} \subseteq (V(H_i) \setminus N^3)$. By Property 7, there is at least one H_i-universal vertex in $V(H_i)$. Since $U_i \cap N^3 = \emptyset$, it must be $U_i \cap N^4 \neq \emptyset$. Assume, without loss of generality, that $z \in U_i \cap N^4$, which implies $z - z'$. By Property 7, $\mathcal{H}$ is not empty, hence, let $H_1 \in \mathcal{H}$. Let $y \in U'_1 \cap N^3$, let x be a neighbor of y in N^2 and let w be a common neighbor of v and y in N^1. Then $\{v, w, x, y, z, z'\}$

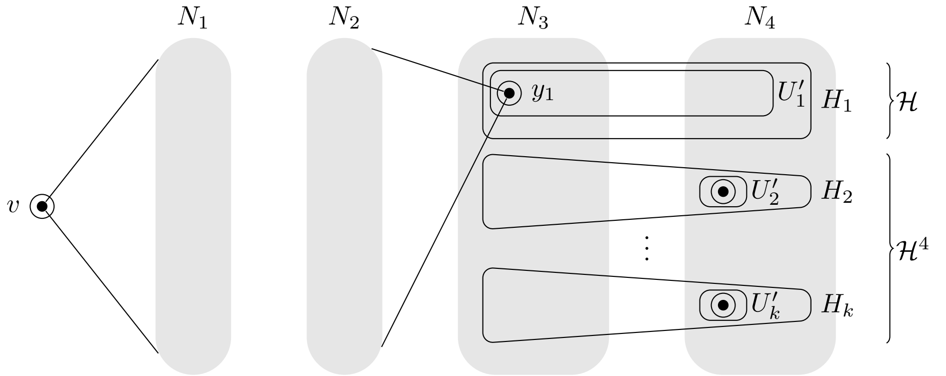

Figure 2.1: Case 1.

induces a $(P_4 + P_2)$ in G because y lies in a different connected component of $G[R]$ than z and z'—this is a contradiction to G's $(P_4 + P_2)$-freeness. □

Lemmas 24 and 25 imply that every v-efficient dominating set D of a $(P_4 + P_2)$-free graph G contains the unique vertex in $V(H_i) \cap N^4$ for every connected component $H_i \in \mathcal{H}^4$. Hence, it remains to analyze how the vertices of the connected components of $\mathcal{H}$ can be chosen into D. We split this analysis into five cases depending on the size of $\mathcal{H}$, the existence and the number of connected components in $\mathcal{H}$ that have no universal vertex in N^4, and, if this number is greater than one, the existence of a vertex in N^2 that is U'_i-universal for some $H_i \in \mathcal{H}$.

The five cases are depicted in the Figures 2.1 to 2.5. Every figure shows a schematic view on the hanging for the vertex v. The distance levels are depicted by grey boxes and the connected components of $G[R]$ as well as the sets $U'_1, \ldots, U'_k$ are drawn as framed boxes. The circled vertices form a valid efficient dominating set. To emphasize the nonexisting of an edge, a dashed line is drawn.

Case 1

Figure 2.1 shows a sketch for Case 1.

Property 10. *There is exactly one N^2-universal vertex y_1 in U'_1.*

Property 11. *For the N^2-universal vertex $y_1 \in U'_1$, we have $D \cap V(H_1) = \{y_1\}$.*

Lemma 26 (Case 1)**.** *If $|\mathcal{H}| = 1$ and G fulfills Properties 7 and 9, then D is a v-efficient dominating set of G, if and only if G fulfills Property 10 and D fulfills Properties 8 and 11.*

Proof. Let D be a v-efficient dominating set of G. By Lemma 24, D fulfills Property 8. Hence, U_1' is the only set that can contain a vertex of D that dominates the vertices of N^2. Since $|D \cap U_1'| = 1$, there must be a vertex $y \in U_1'$ that is N^2-universal. Assume that there is another vertex $y' \in U_1'$ that is N^2-universal. Since both, y and y', are H_1-universal and N^2-universal and have no other neighbors, $\{y, y'\}$ is a homogeneous set of G—this is a contradiction to the primality of G. Hence, y is the only N^2-universal vertex of U_1' and $D \cap V(H_1) = \{y\}$.

Conversely, let G fulfill Property 10 and let D fulfill Properties 8 and 11. Since D contains v, all vertices in N^1 are dominated and, since D fulfills Property 8, all vertices in R are dominated. By Property 11, D also dominates N^2. Every vertex of D, except v, lies in a different connected component of $G[R]$. Since $|\mathcal{H}| = 1$, every vertex of $D \cap R$, except the one of U_1', is in N^4. Hence, all vertices in D have distance at least 3 to each other. Thus, D is efficient and dominating. □

The following property is needed for the cases 2 to 5.

Property 12. *For every $x \in N^2$ and every $H_i \in \mathcal{H}$, there is at most one $y \in U_i'$ with $x \cdots y$. For every $H_i \in (\mathcal{H} \setminus \mathcal{H}^3)$, we have $|V(H_i) \cap N^4| = 1$ and there is exactly one vertex in $U_i' \cap N^3$ and this vertex is N^2-universal.*

Lemma 27. *If $|\mathcal{H}| > 1$ and G is $(P_4 + P_2)$-free, then G fulfills Property 12.*

Proof. Let $x \in N^2$ and let $H_i \in \mathcal{H}$. Assume that x has a non-neighbor $u_i \in U_i'$ and another non-neighbor $y_i \in V(H_i)$. Since u_i is H_i-universal, we have $u_i - y_i$. Since x has non-neighbors in U_i', by definition of U_i', it cannot be U_i-dependent. Hence, x has a neighbor $y_j \in U_j'$ for some $H_j \in \mathcal{H}$ with $i \neq j$. Since y_j is in a different connected component of $G[R]$ than u_i and y_i, a $(P_4 + P_2)$ is induced in G by $\{v, w, x, u_i, y_i, y_j\}$—this is a contradiction.

This shows that every vertex $x \in N^2$ that has a non-neighbor u_i in U_i' for some $H_i \in \mathcal{H}$, is adjacent to every vertex of $V(H_i) \setminus \{u_i\}$. In particular, x is adjacent to every vertex of $U_i' \setminus \{u_i\}$, which proves the first statement of Property 12.

For the second statement, let $H_i \in (\mathcal{H} \setminus \mathcal{H}^3)$ and let $z_i \in U_i' \cap N_4$. Assume that $V(H_i) \cap N^4$ contains another vertex z_i'. Since both, z_i and z_i' are non-adjacent to all vertices of N^2, this is a contradiction to the fact we just proved.

Hence, for $H_i \in (\mathcal{H} \setminus \mathcal{H}^3)$, all vertices of $U_i' \cap N^3$ are N^2-universal and thus, have the same neighborhood in G. Thereby, the primality of G implies that $|U_i' \cap N^3| = 1$. □

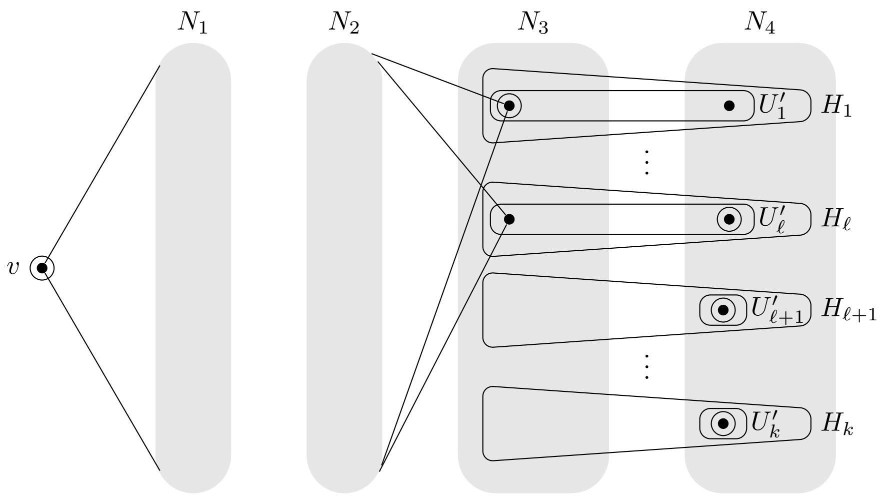

Figure 2.2: Case 2.

Case 2

Figure 2.2 shows a sketch for Case 2.

Property 13. *There is an $H_i \in \mathcal{H}$ such that $D \cap V(H_i) = U'_i \cap N^3$ and for every $H_j \in \mathcal{H}$ with $i \neq j$, we have $D \cap V(H_j) = U'_i \cap N^4$.*

Lemma 28 (Case 2)**.** *If $|\mathcal{H}| > 1$, $\mathcal{H}^3 = \emptyset$, and G fulfills Properties 7 and 12, then D is a v-efficient dominating set of G, if and only if D fulfills Properties 8 and 13.*

Proof. For the first direction, let D be a v-efficient dominating set of G. By Lemma 24, D fulfills Property 8. Since D dominates the vertices of N^2, there must be $y \in N^3 \cap D$. By Property 8, it must be $y \in U'_i$ for some $H_i \in \mathcal{H}$. Since y is N^2-universal and unique by Property 12, we have $D \cap V(H_i) = \{y\}$ and all vertices of N^2 and H_i are dominated. Hence, to dominate the remaining vertices of $G[R]$, there must be vertices of N^4 in D. By Property 12, this must be the unique vertex in $V(H_j) \cap N^4$ for every $H_j \in \mathcal{H}$ with $i \neq j$.

For the other direction, let D fulfill Properties 8 and 13. Since v has distance at least 3 to vertices of R, and the vertices of $R \cap D$ are in different connected components of $G[R]$, and only one vertex of $R \cap D$ is in N^3 and all other vertices in N^4, clearly, D is efficient. Since v dominates N^1, the vertex in $D \cap N^3$ dominates N^2 by Property 12, and every connected component H_i of $G[R]$ is dominated by a vertex of U'_i by Property 8, D is dominating. $\square$

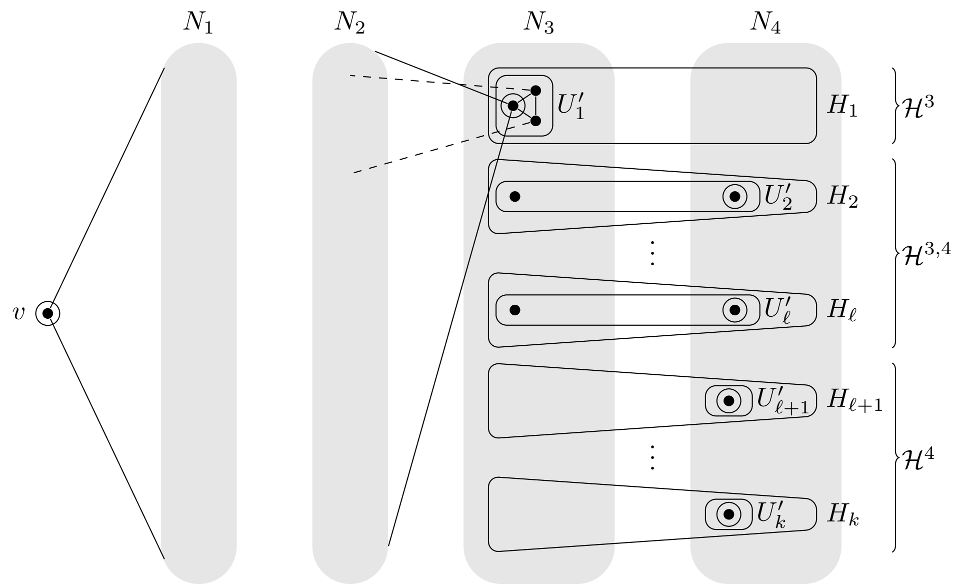

Figure 2.3: Case 3.

Case 3

Figure 2.3 shows a sketch for Case 3.

Property 14. *There is exactly one N^2-universal vertex in U'_1.*

Property 15. *For the N^2-universal vertex $y \in U'_1$, we have $D \cap V(H_1) = \{y\}$. For every $H_i \in \mathcal{H}$ with $i > 1$, we have $D \cup V(H_i) = U'_i \cap N^4$.*

Lemma 29 (Case 3)**.** *If $|\mathcal{H}| > 1$, $|\mathcal{H}^3| = 1$, and G and D fulfill Properties 7 and 12, then D is a v-efficient dominating set of G, if and only if G fulfills Property 14 and D fulfills Properties 8 and 15.*

Proof. Let D be a v-efficient dominating set of G. By Lemma 24, D fulfills Property 8. Since $|\mathcal{H}^3| = 1$, we have $U'_1 \cap N^4 = \emptyset$, hence, $U'_1 \subseteq N^3$. By Property 8, we know that there is $y_1 \in D \cap U'_1$. Let $x \in N^2$ with $x - y_1$. Since $|\mathcal{H}^3| = 1$, every U'_i with $i > 1$ contains a vertex of N^4. Hence, by Property 12, we know that x is adjacent to every vertex of $U'_i \cap N^3$ for every $H_i \in \mathcal{H}$ with $i > 1$. This implies that $D \cap V(H_i) = V(U'_i) \cap N^4$, for every $H_i \in \mathcal{H}$ with $i > 1$ because D is efficient. Since D is dominating, y_1 must be N^2-universal. The existence of another N^2-universal vertex in U'_1 contradicts the primality of G.

Conversely, let G fulfill Property 14 and let D fulfill Properties 8 and 15. Since D contains v, it dominates N^1 and, since it fulfills Property 8, it

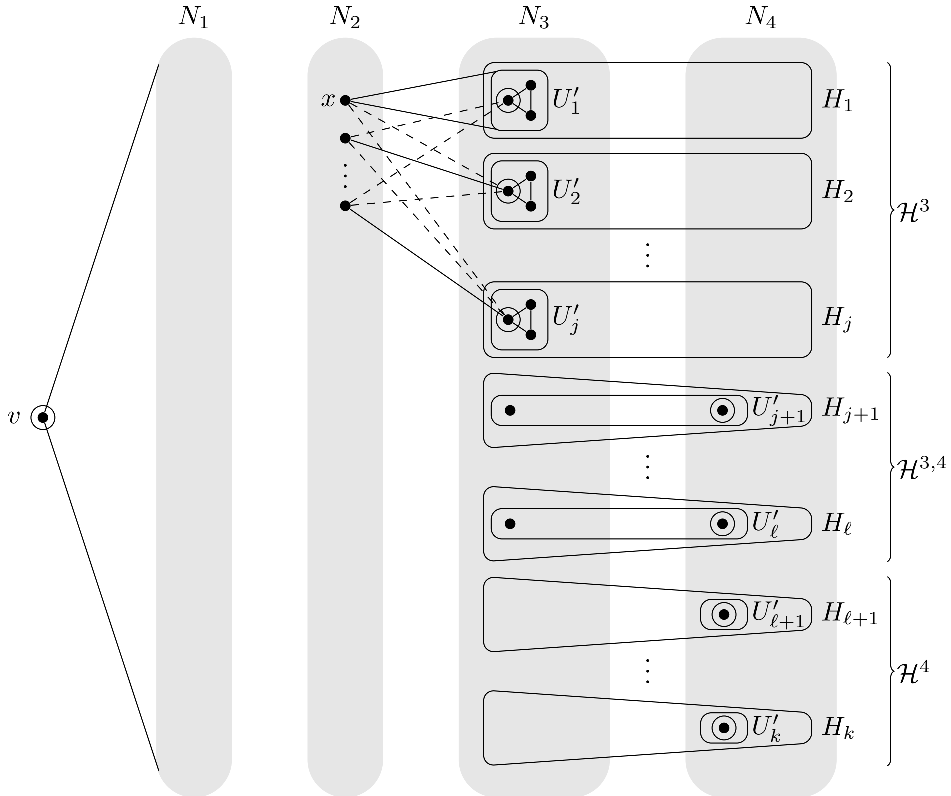

Figure 2.4: Case 4.

dominates R. By Property 15, D contains an N^2-universal vertex $y \in U'_1$ and, hence, it also dominates N^2. Every vertex of D, except v, lies in a different connected component of $G[R]$. Every vertex of $D \cap R$, except y, is in N^4. Hence, all vertices in D have distance at least 3 to each other. Thus, D is efficient and dominating. □

Case 4

Figure 2.4 shows a sketch for Case 4.

Property 16. *Let $x \in N^2$ be a U'_i-universal for some $H_i \in \mathcal{H}$. For every $H_j \in \mathcal{H}$ with $i \neq j$, there is exactly one $y_j \in U'_j$ with $x \cdots y_j$. There is exactly one vertex $y_i \in U'_i$ such that every vertex $x' \in N^2$ is adjacent to y_j for exactly one $H_j \in \mathcal{H}$.*

Property 17. *For every $H_j \in \mathcal{H}$ and y_i as defined in Property 16, we have $D \cap V(H_j) = \{y_j\}$.*

Lemma 30 (Case 4)**.** *If $|\mathcal{H}| > 1$, $|\mathcal{H}^3| > 1$, there is a vertex $x \in N^2$ that is U'_i-universal for some $H_i \in \mathcal{H}$, and G fulfills Properties 7 and 12, then D is a v-efficient dominating set of G, if and only if G fulfills Property 16 and D fulfills Properties 8 and 17.*

Proof. Let D be a v-efficient dominating set of G and let x be a U'_i-universal vertex in N^2 for some $H_i \in \mathcal{H}$. By Lemma 24, D fulfills Property 8. By Property 8, we have $|U'_i \cap D| = 1$. Hence, x is dominated by a vertex $y_i \in U'_i$. Hence, it is adjacent to no other vertex of D. Since, by Property 8, every U'_j contains exactly one vertex of D and, by Property 12, x has at most one non-neighbor in every U'_j, it follows that every U_j with $i \neq j$ contains a vertex y_j with $x \cdots y_j$ and $V(H_j) \cap D = \{y_j\}$. Since D is efficient, dominating, and contains no vertices of N^2, every vertex $x' \in N^2$ must be adjacent to y_j for exactly one $H_j \in \mathcal{H}$. That way, the vertex y_i is uniquely specified: The existence of a vertex $y'_i \in U'_i$ that has exactly the same neighbors in N^2 as y_1 contradicts the primality of G.

Conversely, let G fulfill Property 16 and let D fulfill Properties 8 and 17. Clearly, D dominates N^1 and, by Property 8, also R. By Property 17, D also dominates N^2 because every $x \in N^2$ is adjacent to y_i for exactly one $H_i \in \mathcal{H}$. This also implies that every path between two vertices of D that uses a vertex of N^2 has length at least 3. Since all vertices in $D \cap R$ are in different connected components of $G[R]$, this means that D is efficient and dominating. □

Case 5

Figure 2.5 shows a sketch for Case 5.

Property 18. *There are vertices $\{x, x'\} \subseteq N^2$, $\{a, b\} \subseteq U'_1$, and $\{c, d\} \subseteq U'_2$ such that*

- *$x - a$, $x - c$, $x \cdots b$, $x \cdots d$ and*
- *$x' - b$, $x' - d$, $x' \cdots a$, $x' \cdots c$*

and, if x, x', a, b, c, d are vertices of this kind, then either for $y_1 = a$ and $y_2 = d$ or for $y_1 = b$ and $y_2 = c$:

- *for every $H_i \in \mathcal{H}$ with $i > 2$, there is exactly one vertex $y_i \in U'_i$ with $x \cdots y_i$ and $x' \cdots y_i$ and*
- *for every $x'' \in N^2$, there is exactly one $H_i \in \mathcal{H}$ such that $x'' - y_i$.*

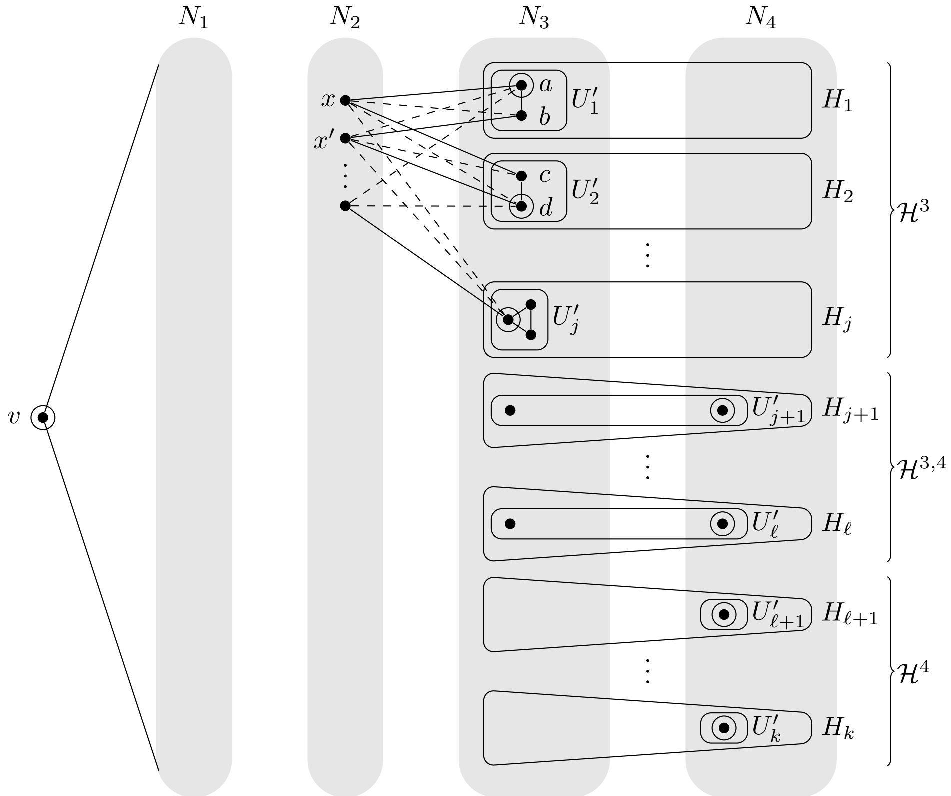

Figure 2.5: Case 5.

Property 19. *For every $H_i \in \mathcal{H}$ and y_i as defined in Property 18, we have $D \cap V(H_i) = \{y_i\}$.*

Lemma 31 (Case 5)**.** *If $|\mathcal{H}| > 1$, $|\mathcal{H}^3| > 1$, there is no vertex $x \in N^2$ that is U'_i-universal for some $H_i \in \mathcal{H}$, and G fulfills Properties 7 and 12, then D is a v-efficient dominating set of G, if and only if G fulfills Property 18 and D fulfills Properties 8 and 19.*

Proof. The existence of x, x', a, b, c, and d follows from $|\mathcal{H}^3| > 1$ and the fact that there is no U'_1- or U'_2-universal vertex in N^2: Let $w \in N^2$. Since w is not U'_1-universal, there is $y'_1 \in U'_1$ with $w \cdots y'_1$. Let w' be a neighbor of y'_1 in N^2. Since w' is not U'_1-universal, there is a vertex $y_1 \in U'_1$ with $w' \cdots y_1$. Clearly, we have $w - y_1$ because, by Property 12, w has at most one non-neighbor in U'_1. Since w and w' are not U'_2-universal, there are vertices $y_2 \in U'_2$ and $y'_2 \in U'_2$ with $w \cdots y'_2$ and $w' \cdots y_2$. Notice that it may be $y_2 = y'_2$. If this is not the case, then $x := w$, $x' := w'$, $a := y_1$, $b := y'_1$, $c := y_2$, and $d := y'_2$ fulfill the desired properties. Hence, assume that $y_2 = y'_2$, that means y_2 is a common non-neighbor of w and w'. Let $w'' \in N^2$ with $w''-y_2$. Since w'' is neither U'_1-universal nor U'_2-universal, there are $y''_1 \in U'_1$ and $y''_2 \in U'_2$ with $w'' \cdots y''_1$ and $w'' \cdots y''_2$. It may be $y''_1 = y_1$ or $y''_1 = y'_1$. It is easy to check that, in any case, either

$$x := w, x' := w'', a := y_1, b := y'_1, c := y''_2, \text{ and } d := y_2 \text{ or}$$
$$x := w', x' := w'', a := y'_1, b := y_1, c := y''_2, \text{ and } d := y_2$$

fulfill the desired properties.

Let D be a v-efficient dominating set of G. By Lemma 24, D fulfills Property 8. Assume that $\{a, b\} \cap D = \emptyset$. Since, by Property 12, x and x' have at most one non-neighbor in U'_1 and, by Property 8, U'_1 contains one vertex of D, x is dominated by a vertex $y \in U'_1$ that also dominates x'. By the existence of c and d and Property 12, every vertex of U'_2 is adjacent to x or x' or both. By Property 8, U'_2 also contains a vertex of D. Hence, either x or x' or both have two neighbors in D—this is a contradiction to the efficiency of D. This implies that either $a \in D$ or $b \in D$.

Assume that $a \in D$. Since, by Property 8, U'_2 contains a vertex of D and d is the only non-neighbor of a in U'_2, we have $d \in D$. Let $y_1 := a$ and $y_2 := d$. By Property 8, every U'_i must contain a vertex of D. Hence, since x and x' are dominated by y_1 and y_2 and D is efficient, there is a vertex $y_i \in U'_i$ with $x \cdots y_i$, $x' \cdots y_i$, and $D \cap V(H_i) = \{y_i\}$ for every $H_i \in \mathcal{H}$ with $i > 2$. Clearly, since D is efficient, dominating, and contains no vertex of N^2, every vertex $x \in N^2$ is adjacent to y_i for exactly one $H_i \in \mathcal{H}$. The proof works analogously if $b \in D$ with $y_1 := b$ and $y_2 := c$.

Conversely, assume that G fulfills Property 18 and D fulfills Properties 8 and 19. Clearly, D dominates N^1. By Property 8, D also dominates R. By Property 19, every vertex of N^2 has exactly one neighbor in D and, hence, N^2 is dominated as well and every path between two vertices of D that uses a vertex of N^2 has length at least 3. Since all vertices in $D \cap R$ are in different connected components of $G[R]$, this means that D is efficient and dominating. □

These lemmas lead to the procedure shown in Tables 2.3 and 2.4 that robustly works on $(P_4 + P_2)$-free graphs.

Lemma 32. *The procedure* `Robust-`(P_4+P_2)`-free-`v`-MWED` *is correct in the sense of Definition 2 and runs in time* $O(|V| + |E|)$.

Proof. The correctness follows from Lemmas 23 to 31 because the algorithm stops, if it finds an induced $(P_4 + P_2)$ in G or if G cannot have a v-efficient dominating set, and it guarantees, by Step 7 and one of the Steps 8.1, 9.2.3, 9.3.2, 9.4.1.2, and 9.4.2.5, that D fulfills Property 8 and one of the Properties 11, 13, 15, 17, and 19.

The rest of the proof shows the time bound. First, we have to show that Step 4 can be implemented in time $O(|V|+|E|)$, hence, that we can determine the connected components $H_1, \ldots, H_k$ of $G[R]$, the sets $\mathcal{H}^3, \mathcal{H}, \mathcal{H}^4$, and U_i' for every H_i in that time. Since $\mathcal{N}_v$ is part of the input, we can assume that every vertex $v \in N^i$ is labeled with $\ell(N^i)$.

Clearly, the connected components of $G[R]$ can be found in linear time by breadth-first-search or depth-first-search. We can assume that every vertex y of R has the label $\ell(H_i)$, if and only if $y \in V(H_i)$. For every H_i, the set U_i of H_i-universal vertices of H_i can be found by counting the degree of every vertex in H_i. The vertices with degree $|V(H_i)|$ are exactly the vertices in U_i, we label these vertices with $\ell(U_i)$. The degree can be counted in time $O(|E|)$ by starting with degree 0 for every vertex, considering every edge, checking if both endpoints have the same label, say $\ell(H_i)$, and, if so, increasing the degree of both endpoints.

For every H_i, we can decide if $H_i \in \mathcal{H}^3$, $H_i \in \mathcal{H}$ or $H_i \in \mathcal{H}^4$ by checking if every vertex of U_i is labeled $\ell(N^3)$, at least one is labeled $\ell(N^3)$, or none is labeled $\ell(N^3)$ respectively.

Procedure: `Robust-`(P_4+P_2)`-free-`v`-MWED(`$G=(V,E)$`, `v`, `$\mathcal{N}_v$`)`
Input: As defined in Definition 2.
Output: As defined in Definition 2.

1. Let `STOP-1` := `STOP:` G `is not` (P_4+P_2)`-free`.
2. Let `STOP-2` := `STOP:` G `admits no` v`-efficient dominating set`.
3. Set $D := \{v\}$.
4. Determine $H_1, \ldots, H_k$, $\mathcal{H}$, $\mathcal{H}^4$, $\mathcal{H}^3$, and U_i' for every $i \in \{1, \ldots, k\}$.
5. Check if G fulfills Property 6. If not, then `STOP-1`.
6. Check if G fulfills Property 7. If not, then `STOP-2`.
7. Check if G fulfills Property 9. If so, set $D := D \cup U_i'$ for every $H_i \in \mathcal{H}^4$, otherwise `STOP-1`.

Table 2.3: The procedure `Robust-`(P_4+P_2)`-free-`v`-MWED` (part 1/2)

8. If $|\mathcal{H}| = 1$: /* *Case 1* */
   1. Check if G fulfills Property 10. If so, set $D := D \cup \{y_1\}$ for the N^2-universal vertex $y_1 \in U'_1$, otherwise `STOP-2`.

   — *G and D fulfill Properties 6 to 11.* —
9. If $|\mathcal{H}| > 1$:
   1. Check if G fulfills Property 12. If not, then `STOP-1`.
   2. If $\mathcal{H}^3 = \emptyset$: /* *Case 2* */
      1. For every $H_i \in \mathcal{H}$, let y_i be the vertex in $U'_i \cap N^3$ and z_i the vertex in $U'_i \cap N^4$.
      2. Find an $H_i \in \mathcal{H}$ such that $\omega(y_i) + \sum_{H_j \in \mathcal{H}, i \neq j} \omega(z_j)$ is minimized.
      3. Set $D := D \cup \{y_i\} \cup \{z_j \mid H_j \in \mathcal{H}, i \neq j\}$.

      — *G and D fulfill Properties 6 to 9 and 13.* —
   3. If $\mathcal{H}^3 = \{H_1\}$: /* *Case 3* */
      1. Check if G fulfills Property 14. If not, then `STOP-2`.
      2. Set $D := D \cup \{y_1\} \cup \{y_i \mid i > 1, H_i \in \mathcal{H}, \{y_i\} = U'_i \cap N^4\}$.

      — *G and D fulfill Properties 6 to 9, 14, and 15.* —
   4. If $|\mathcal{H}^3| > 1$: /* *Case 4 and 5* */
      1. If N^2 contains a vertex x that is U'_i-universal for some $H_i \in \mathcal{H}$:
         1. Check if G fulfills Property 16. If not, then `STOP-2`.
         2. Set $D := D \cup \{y_i \mid H_i \in \mathcal{H}\}$ for y_i as defined in Property 16.

         — *G and D fulfill Properties 6 to 9, 16, and 17.* —
      2. If N^2 contains no vertex that is U'_i-universal for some $H_i \in \mathcal{H}$:
         1. Check if G fulfills Property 18. If not, then `STOP-2`.
         2. Find vertices x, x', a, b, c, d as defined in Property 18.
         3. If $\omega(a) + \omega(d) \leq \omega(b) + \omega(c)$, then set $y_1 := a; y_2 := b$, otherwise set $y_1 := b; y_2 := c$.
         4. For every $H_i \in \mathcal{H}$ with $i > 2$ find vertex y_i as defined in Property 18.
         5. Set $D := D \cup \{y_i \mid H_i \in \mathcal{H}\}$.

         — *G and D fulfill Properties 6 to 9, 18, and 19.* —
10. `Return` D.

Table 2.4: The procedure `Robust-`$(P_4 + P_2)$`-free-`v`-MWED` (part 2/2)

For every $y \in N^3$, let $n^2(y) = |N(y) \cap N^2|$. This can be computed in time $O(|V| + |E|)$ by starting with $n^2(y) = 0$ for all $y \in N^3$, considering every edge xy, checking if x is labeled $\ell(N^2)$ and y is labeled $\ell(U_i)$ for some i, and, if so, increasing $n^2(y)$ by 1.

A vertex $x \in N^2$ is U_i-dependent, if $N(x) \cap U_i = \emptyset$ for all i but at most one. This can be checked in time $O(|V| + |E|)$ in the following way, using the new labels $\ell(Z), \ell(M), \ell(D_1), \ldots, \ell(D_k)$: Assign the label $\ell(Z)$ to every vertex $x \in N^2$. Consider every edge xy with x labeled $\ell(N^2)$ and y labeled $\ell(U_i)$. If x is labeled $\ell(Z)$, then remove this label and add the label $\ell(D_i)$. If x is labeled $\ell(D_j)$ for some $i \neq j$, then remove this label and add the label $\ell(M)$. If x is labeled $\ell(D_i)$, do nothing. Clearly, after all edges are considered, a vertex $x \in N^2$ is labeled $\ell(Z)$, if it has no neighbor in any U_i, it is labeled $\ell(D_i)$, if it has no neighbors in any U_j but U_i, and it is labeled $\ell(M)$, if it has neighbors in U_i and U_j for at least one pair $i \neq j$. Notice that, if a vertex $x \in N^2$ is labeled with $\ell(D_i)$, then it is not labeled with $\ell(D_j)$ for all $i \neq j$. Hence, x is U_i-dependent, if and only if it is labeled $\ell(D_i)$.

Next, we have to construct U'_i from U_i for every i. Therefore, let $d_U(U_i) := |D_i|$ for all U_i and, for every U_i and every $y \in U_i$, let $d(y) := |N(y) \cap D_i|$. These values can be computed in time $O(|V| + |E|)$ in the following way, starting with $d_U(U_i) = 0$ and $d(y) = 0$ for all U_i and for all $y \in U_i$: Firstly, consider every vertex $x \in N^2$ and, if x is labeled $\ell(D_i)$, then increase $d_U(U_i)$ by 1. Secondly, consider every edge xy and, if x is labeled $\ell(D_i)$ and y is labeled $\ell(U_i)$, then increase $d(y)$ by 1. Clearly, U'_i contains a vertex $y \in U_i$, if and only if either $d_U(U_i) = 0$ or $d(y) = d_U(U_i)$. Hence, the sets U'_i can be built in time $O(|V| + |E|)$. For every i, we label every vertex of U'_i with $\ell(U'_i)$.

Property 6 can be checked in time $O(|V| + |E|)$ because recognizing cographs can be done in linear time [30]. Clearly, Property 7 can be checked during building the sets U'_i. Additionally, we determine the partition $\mathcal{H} = \mathcal{H}^{=1} \cup \mathcal{H}^{>1}$ alongside, where $H_i \in \mathcal{H}$ is in $\mathcal{H}^{=1}$, if and only if $|U'_i| = 1$.

We can check Property 9 by considering every $H_i \in \mathcal{H}$ and check if exactly one $y \in V(H_i)$ has the label $\ell(N^4)$. This takes at most $O(|R|)$ time.

In case 1, we consider every vertex of $y \in U'_1$ and check if $n^2(y) = |N^2|$ to test Property 10. This takes at most $O(|R|)$ time.

If every vertex $x \in N^2$ has at most one non-neighbor in every set U'_i, then x has at least $|\mathcal{H}^{>1}|$ neighbors in N^3. Hence, we have $|E| \geq |N^2| \cdot |\mathcal{H}^{>1}|$. To test Property 12 in time $O(|V| + |E|)$, we can do the following: Check if $|E| \geq |N^2| \cdot |\mathcal{H}^{>1}|$. If this check fails, G violates the property. Otherwise, determine $n_i^3(x) = |N(x) \cap U'_i|$ for every $x \in N^2$ and $H_i \in \mathcal{H}^{>1}$. This can be done by starting with $n_i^3(x) = 0$ for every $x \in N^2$ and $H_i \in \mathcal{H}^{>1}$, considering every edge, checking if one endpoint of the edge is labeled $\ell(N^2)$ and the other endpoint is labeled $\ell(U_i')$ for some $H_i \in \mathcal{H}^{>1}$, and, if so, increasing $n_i^3(x)$ by 1. Since there are $|N^2| \cdot |\mathcal{H}^{>1}|$ values to determine and $|N^2| \cdot |\mathcal{H}^{>1}| \leq |E|$, this can be done in time $O(|E|)$. Then, check for every $x \in N^2$ and every $H_i \in \mathcal{H}^{>1}$, if $n_i^3(x) \geq |U_i'| - 1$. If this is true, then the first part of Property 12 is fulfilled because every vertex has at most one non-neighbor in U_i' for every $H_i \in \mathcal{H}^{=1}$. To check the second part of Property 12, we consider every $H_i \in (\mathcal{H} \setminus \mathcal{H}^3)$, check if at most one vertex of $V(H_i)$, say y', is labeled with $\ell(N^4)$, check if y' is also labeled with $\ell(U_i')$ and if there is exactly one $y \in V(H_i)$ that is different from y' and labeled with $\ell(U_i')$, check if y is labeled $\ell(N^3)$, and check if $n^2(y) = |N^2|$. Clearly, if and only if all checks succeed, then G fulfills Property 12.

For case 2, the vertices y_i and z_i can be found while checking Property 12. We calculate

$$s_i := \omega(y_i) + \sum_{H_j \in \mathcal{H}, i \neq j} \omega(z_j)$$

for every $H_i \in \mathcal{H}$ in the following way: Calculate s_i for a fixed i with $H_i \in \mathcal{H}$ as the formula implies. This takes at most $O(|V|)$ time. To get s_j for $H_j \in \mathcal{H}$ with $i \neq j$, we simply compute $s_j := s_i - \omega(y_i) + \omega(y_j) + \omega(z_i) - \omega(z_j)$. Hence, we can calculate all s_i in time $O(|V|)$ and find an $H_i \in \mathcal{H}$ such that s_i is minimized.

In case 3, Property 14 can simply be checked by considering $n^2(y)$ for every $\ell(U_1')$-labeled $y \in N^3$. The vertices y_i with $i > 1$, $H_i \in \mathcal{H}$ and $\{y_i\} = U_i' \cap N^4$ are already found when Property 12 was tested, so we can reuse them.

For case 4 and 5 we have to check if there is a vertex $x \in N^2$ that is U_i'-universal for some $H_i \in \mathcal{H}$. This is the case, if $\mathcal{H}^{=1} \neq \emptyset$ or $n_i^3(x) = |U_i'|$ for some $x \in N^2$ and $H_i \in \mathcal{H}^{>1}$. If there is an $H_i \in \mathcal{H}^{=1}$, then the neighbor x of y_i in N^2 for $\{y_i\} = U_i'$ is U_i'-universal. This can be tested in constant time. If there is an $x \in N^2$ and an $H_i \in \mathcal{H}^{>1}$ with $n_i^3(x) = |U_i'|$, then x is U_i'-universal by definition. Since all these values are already computed and there are at most $|E|$ of these values, this can be tested in time $O(|E|)$.

Property 16 can be checked in time $O(|V| + |E|)$ in the following way: Let $x \in N^2$ be U_i'-universal, as found in the previous step. Consider every edge, check if one of its endpoints equals x and, if so, label the other endpoint with $\ell(x)$. Then check for every $H_j \in \mathcal{H}$ with $i \neq j$, if U_i' contains exactly one vertex y_j that has not the label $\ell(x)$. This check can be done for all $H_i \in \mathcal{H}$ in time $O(|R|)$ and, if it fails, then G violates Property 16. Give every such y_j the label $\ell(y_j)$. Now pick a neighbor $x'' \in N^2$ of some y_j with $H_j \in \mathcal{H}$ and

$i \neq j$ and find $y_i \in U_i'$ with $x'' \cdots y_i$. This can be done in time $O(|V| + |E|)$ by considering every edge and using a label $\ell(x'')$ like above. If it does not exist, G violates Property 16. Give y_i the label $\ell(y_i)$. Now consider every edge and check if one endpoint, say y, is labeled with $\ell(y_j)$ for some $H_j \in \mathcal{H}$ and the other endpoint, say x, is labeled with $\ell(N^2)$. If so, label x with $\ell(L)$. If x already has this label, then G violates Property 16 because there is a vertex $x' \in N^2$ that has two neighbors y_j and y_k for $\{H_j, H_k\} \subseteq \mathcal{H}$ and $j \neq k$. After that, consider every vertex of N^2 and check if it is labeled with $\ell(L)$. If this is not the case for a vertex x, then G violates Property 16 because x is not adjacent to y_j for all $H_j \in \mathcal{H}$. If all checks succeed, then G clearly fulfills Property 16.

The proof of Lemma 31 describes a way to find suitable vertices x, x', a, b, c, and d that fulfill Property 18, if such vertices exist: Choose $w \in N^2$ arbitrarily and find a non-neighbor $y_1' \in N^2$ of w. Then find a neighbor $w' \in N^2$ of y_1' and a non-neighbor $y_1 \in U_1'$ of w'. Finally, find a non-neighbor $y_2' \in U_2'$ of w and a non-neighbor $y_2 \in U_2'$ of w'. This can be done in time $O(|V| + |E|)$ because to find a neighbor or a non-neighbor of a given vertex in a labeled set, we can consider every edge and a labelling technique that is similar to the techniques used above. If $y_2 = y_2'$, find a neighbor $w'' \in N^2$ of y_2 and two non-neighbors, $y_1'' \in U_1'$ and $y_2'' \in U_2'$ of w''. The existence of all these vertices is guaranteed because we already know that G fulfills Property 12 and the fact that there is no U_i'-universal vertex in N^2 for any U_i, hence, every U_i' contains at least two vertices. Finally, test in constant time if there is an assignment for $\{x, x'\} \subseteq \{w, w, w''\}$ and $\{a, b, c, d\} \subseteq \{y_1, y_1', y_1'', y_2, y_2', y_2''\}$ such that $x{-}a$, $x{-}c$, $x'{-}b$, $x'{-}d$, $x \cdots b$, $x \cdots d$, $x' \cdots a$, and $x' \cdots c$. If this is not possible, then G violates Property 18. Otherwise, find a common non-neighbor $y_i \in U_i'$ of x and x' for every $H_i \in \mathcal{H}$ with $i \geq 2$. This can also be done using labels in time $O(|V| + |E|)$. If at least one of these non-neighbors does not exist, then G violates Property 18. Finally, check for every $x'' \in N^2$, if it is adjacent to y_i for exactly one $H_i \in \mathcal{H}$. This can be done in time $O(|V| + |E|)$ in the same way we check the last part of Property 16. If all checks succeed, then G clearly fulfills Property 18. □

Together with Lemma 17, Lemma 32 implies:

Theorem 8. MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in time* $O(nm)$ *on* $(P_4 + P_2)$*-free graphs in a robust way, where* n *is the number of vertices and* m *is the number of edges of the input graph.*

2.3 Reduction from Efficient Domination to Maximum Weighted Independent Set

MAXIMUM INDEPENDENT SET is one of the classical NP-complete graph problems. The weighted optimization version can be formulated as follows:

Definition 3. *Given a graph $G = (V, E)$ with vertex weights ω and a constant k,* MAXIMUM WEIGHT INDEPENDENT SET *asks if G admits an independent set $I \subseteq V$ with $\omega(I) \geq k$.*

Although MAXIMUM WEIGHT INDEPENDENT SET remains NP-complete on several restricted graph classes, for example triangle-free graphs [84], $K_{1,4}$-free graphs [77], and planar graphs of maximum degree at most 3 [48], there are lots of tractability results, for example for claw-free graphs [77, 80, 82] and their generalization fork-free graphs [69], for perfect graphs [50], and many others.

The *square* G^2 of a graph G has the same vertex set as G and two vertices x and y are adjacent, if and only if $\text{dist}_G(x, y) \leq 2$. For a graph class $\mathcal{C}$, we denote by $\mathcal{C}^2 := \{G^2 \mid G \in \mathcal{C}\}$ the graph class whose graphs are the squares of the graphs of $\mathcal{C}$. Let α be the best known exponent such that matrix multiplication of two $n \times n$-matrices can be done in time $O(n^\alpha)$.

EFFICIENT DOMINATION on G can be reduced to MAXIMUM WEIGHT INDEPENDENT SET on G^2 as follows:

Lemma 33 ([11, 76])**.** *Let $\mathcal{C}$ be a graph class such that* MAXIMUM WEIGHT INDEPENDENT SET *can be solved in time $T(|G|)$ on graphs $G \in \mathcal{C}^2$.* EFFICIENT DOMINATION *is solvable in time $O(\min\{nm + n, n^\alpha\} + T(|G^2|))$ on graph $G \in \mathcal{C}$, where n is the number of vertices and m the number of edges of the input graph.*

The idea of this reduction is to define vertex weights ω in G^2 as $\omega(v) := |N_G[v]|$. Clearly, every independent set I of G^2 coincides with an efficient set in G. If $\omega(I) = |V|$, then I is also dominating in G.

In [7], this is extended to the weighted optimization versions of EFFICIENT DOMINATION:

Theorem 9 (Theorem 3 in [7])**.** *Let $\mathcal{C}$ be a graph class such that* MAXIMUM WEIGHT INDEPENDENT SET *can be solved in time $T(|G|)$ on graphs $G \in \mathcal{C}^2$.* MINIMUM/MAXIMUM WEIGHT EFFICIENT DOMINATION *is solvable in time $O(\min\{nm + n, n^\alpha\} + T(|G^2|))$ on graphs $G \in \mathcal{C}$, where n is the number of vertices and m the number of edges of the input graph.*

This enables the design of an efficient algorithm for MINIMUM WEIGHT EFFICIENT DOMINATION on a graph class $\mathcal{C}$ by proving that $\mathcal{C}^2$ has properties that make MAXIMUM WEIGHT INDEPENDENT SET tractable on $\mathcal{C}^2$.

In this section, we refine this approach: For a graph class $\mathcal{C}$, let $\mathcal{C}^2_{ed}$ contain every efficiently dominatable graph of $\mathcal{C}^2$. We seek for a polynomial-time decidable graph class $\mathcal{D}$ with $\mathcal{C}^2_{ed} \subseteq \mathcal{D}$ that allows efficient solving of MAXIMUM WEIGHT INDEPENDENT SET. For such a graph class $\mathcal{D}$, we can solve MINIMUM WEIGHT EFFICIENT DOMINATION of a graph $G \in \mathcal{C}$ by constructing G^2 and checking if $G^2 \in \mathcal{D}$. If $G^2 \notin \mathcal{D}$, then G admits no efficient dominating set. Otherwise, we use an efficient MAXIMUM WEIGHT INDEPENDENT SET-algorithm for $\mathcal{D}$ to solve MINIMUM WEIGHT EFFICIENT DOMINATION on G, which works by Theorem 9. Corollary 34 summarizes this procedure:

Corollary 34. *Let $\mathcal{C}$ and $\mathcal{D}$ be graph classes such that $\mathcal{D}$ contains the square of every efficiently dominatable graph of $\mathcal{C}$, let $G \in \mathcal{D}$ be decidable in time $T_{\mathcal{D}}(|G|)$ and let* MAXIMUM WEIGHT INDEPENDENT SET *be solvable for $G \in \mathcal{D}$ in time $T(|G|)$. Then* MINIMUM/MAXIMUM WEIGHT EFFICIENT DOMINATION *is solvable in time $O(\min\{nm + n, n^\alpha\} + T_{\mathcal{D}}(|G^2|) + T(|G^2|))$ for $G \in \mathcal{C}$, where n is the number of vertices and m the number of edges of the input graph.*

The next section applies this technique to P_5-free graphs by showing that the squares of P_5-free graphs that admit an efficient dominating set are cographs.

2.3.1 P_5-free Graphs

The following observation simplifies the main proof of this section:

Observation 35. *Let $G = (V, E)$ be a P_5-free graph that has an induced P_4 a–b–c–d. If D is an efficient dominating set of G, then $\{b, c\} \cap D = \emptyset$.*

Proof. Without loss of generality, assume that $b \in D$. Since D dominates d, there is a vertex $d' \in D$ with d–d'. Since D is efficient, we have $d' \notin \{a, b, c\}$ and $a \cdots d'$, $b \cdots d'$, and $c \cdots d'$. Hence, a–b–c–d–d' is an induced P_5 in G—this is a contradiction. $\square$

Theorem 10. *If a graph G is P_5-free and efficiently dominatable, then G^2 is P_4-free, that is, a cograph.*

Proof. Let $G = (V, E)$ be a P_5-free graph and let D be an efficient dominating set of G. Assume that G^2 contains at least one induced P_4. Let $a–b–c–d$ be a P_4 of G with $a \in D$. If such a P_4 does not exist, let $a–b–c–d$ be an arbitrarily chosen P_4. We have

$$\begin{aligned}&\operatorname{dist}_G(a,b) \leq 2,\ \operatorname{dist}_G(b,c) \leq 2,\\&\operatorname{dist}_G(c,d) \leq 2,\ \operatorname{dist}_G(a,c) \geq 3,\\&\operatorname{dist}_G(a,d) \geq 3, \text{ and } \operatorname{dist}_G(b,d) \geq 3\end{aligned}$$

because, otherwise $\{a, b, c, d\}$ does not induce a P_4 in G^2. Let $P \subseteq V$ be a minimal vertex set such that $\{a, b, c, d\}$ induces a P_4 in $G[P]^2$. Clearly, we have $\{a, b, c, d\} \subseteq P$. It is easy to check that the inequalities from above cannot be fulfilled with $P = \{a, b, c, d\}$, thus, P contains some additional vertices. Every vertex of $x \in P \setminus \{a, b, c, d\}$ must be adjacent either to a and b, or b and c, or c and d: If x were adjacent to at most one vertiex of $\{a, b, c, d\}$, then P is not minimal. If x were adjacent to a and c, a and d, b and d, or more than two vertices of $\{a, b, c, d\}$, then $\{a, b, c, d\}$ is not an induced P_4 in G^2. Thus, we have the following cases to analyze:

(1) $P = \{a, b, c, d, x\}$ with $a–b–x–c–d$ and $b \cdots c$,

(2) $P = \{a, b, c, d, x, y\}$ with $a–x–b–y–c–d$, $a \cdots b$, and $b \cdots c$,

(3) $P = \{a, b, c, d, x, y\}$ with $a–x–b–c–y–d$, $a \cdots b$, and $c \cdots d$, and

(4) $P = \{a, b, c, d, x, y, z\}$ with $a–x–b–y–c–z–d$, $a \cdots b$, $b \cdots c$, and $c \cdots d$.

Case (1): It is easy to check that $a–b–x–c–d$ must be induced in G—this is a contradiction to the P_5-freeness of G.

Case (2): We have $x–y$ because otherwise G contains the induced P_5 $a–x–b–y–c$—this is a contradiction.

Case (3): Clearly, $x–y$ because otherwise $a–x–b–c–y–d$ would be induced in G. Since $a–x–b–c$ and $b–c–y–d$ are induced, Observation 35 implies

$$D \cap \{b, c, x, y\} = \emptyset. \tag{2.1}$$

Assume that $a \notin D$. By symmetry and the choice of $a–b–c–d$, we also have $d \notin D$. Since D is dominating, there is $a' \in D$ with $a–a'$. By the inequalities, we have $a' \cdots c$ and $a' \cdots d$. Since $a' \in D$, by Observation 35, $a–a'–b–c$ cannot be induced. Hence, we have $a' \cdots b$. This is also the case for $a–a'–y–c$, so we have $a' \cdots y$. This shows $N(a) \cap P = N(a') \cap P$, what implies that $\{a', b, c, d\}$ induces a P_4 in $G^2[P']$ with $a' \in D$ for $P' = \{a', b, c, d, x, y\}$—this is a contradiction to the choice of a–b–c–d.

Assume that $a \in D$. Since D is dominating, there is a vertex $c' \in D$ with c–c'. The inequalities and Equation (2.1) imply that $c' \notin \{a, b, c, d, x, y\}$. Since D is efficient, we have $c' \cdots x$. The path a–x–b–c–c' cannot be induced, hence, the inequalities imply b–c'. By Observation 35, x–b–c'–d cannot be induced. Hence, we have $c' \cdots d$. The path c'–b–x–y–d' cannot be induced, hence, we have c'–y. The induced P_4 b–c'–y–d with $c' \in D$ is a contradiction to Observation 35.

Case (4): It is easy to check that x–y–z–x because otherwise G would contain an induced P_5. Since a–x–y–c and b–y–z–d are induced, Observation 35 implies

$$D \cap \{x, y, z\} = \emptyset. \tag{2.2}$$

Assume that $a \in D$. Since D is dominating, there is $b' \in D$ with b–b'. Since $b \cdots c$, by the minimality of P, to fulfill the inequalities, it must be $b' \notin \{a, b, c, d, x, y, z\}$ and $b' \cdots d$. The efficiency of D implies that $b' \cdots x$. The induced P_4 b–b'–z–d is a contradiction to Observation 35. Hence, $a \notin D$ and, by symmetry, $d \notin D$.

Assume that $b \in D$. Since D is efficient, by Equation (2.2), we have $D \cap \{a, c, x, y, z\} = \emptyset$. Hence, there is $c' \in D$ with c–c' and, clearly, $c' \notin \{a, b, c, x, y\}$. The inequalities imply that $a \cdots c'$ and the efficiency of D implies that $c' \cdots b$, $c' \cdots x$, and $c' \cdots y$. The path a–x–y–c–c' is induced—this is a contradiction. Hence, $b \notin D$ and, by symmetry, $c \notin D$.

Since D is dominating and $D \cap \{a, b, c, d, x, y, z\} = \emptyset$, there is an $a' \in D$ with a–a'. By the inequalities, we have $a' \cdots c$ and $a' \cdots d$. By Observation 35, a–a'–y–c cannot be induced. Hence, $a' \cdots y$. The P_5 a'–a–x–y–c cannot be induced, thus, we have a'–x. Since $c \notin D$, $a' \cdots c$, and D is dominating, there is a $c' \in D$ with c–c'. By the efficiency of D, we have $a' \cdots c'$, $a \cdots c'$, and $c' \cdots x$. Since a'–x–y–c–c' cannot be induced, it must be c'–y. Observation 35 forbids that d–c'–y–x is an induced P_4, thus, we have $c' \cdots d$. Hence, d is still not dominated. So there is a $d' \in D$ with d–d'. By the efficiency of D, we have $a' \cdots d'$, $a \cdots d'$, $d' \cdots x$, $c' \cdots d'$, $c \cdots d'$, and $d' \cdots y$. Since c'–c–z–d–d' cannot be induced, we have c'–z or d'–z. If c'–z, then, by the efficiency of D, we have $a' \cdots z$ and $d' \cdots z$ and a'–x–z–d–d' is an induced P_5 in G—this is a contradiction. If d'–z, then, by the efficiency of D, we have $a' \cdots z$ and $c' \cdots z$ and a'–x–z–c–c' is an induced P_5 in G—this is a contradiction. □

Theorem 10 allows to apply Corollary 34, where $\mathcal{C}$ is the class of P_5-

free graphs and $\mathcal{D}$ is the class of cographs. Cographs can be recognized in linear time [30, 66] and MAXIMUM WEIGHT INDEPENDENT SET can be solved on cographs in linear time for several reasons, for example because cographs are the graphs with clique-width 2 [32] and MAXIMUM WEIGHT INDEPENDENT SET can be formulated in monadic second oder logic without edge set quantification. Since the $\min\{nm+n, n^\alpha\}$ in the runtime given in Corollary 34 results from the computation of G^2, the minimum is bounded by $|G^2|$. With this, we get:

Corollary 36. MINIMUM WEIGHT EFFICIENT DOMINATION *is solvable in time* $O(\min\{nm, n^\alpha\})$ *on* P_5*-free graphs with* n *vertices and* m *edges.*

As α is known to be near 2.37 [29, 93], this runtime is as good as or even better, in some cases, than the runtime given in Theorem 7 using the hanging technique. The advantage of the algorithm using the hanging technique is its robustness. Since it is not known if P_5-free graphs can be recognized in time $O(nm)$, the technique of Corollary 36 can show its performance only if the input graphs is already known to be P_5-free.

3 Efficient Edge Domination

This chapter presents an $O(n+m)$-time MINIMUM WEIGHT EFFICIENT EDGE DOMINATION algorithm for chordal bipartite graphs and a robust $O(nm)$-time algorithm for hole-free graphs.

The results are achieved by reducing MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on chordal bipartite graphs and hole-free graphs respectively to MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on K_4-free block graphs, a class that admits a linear time algorithm for this problem.

The results are published in the extended abstract [8]. However, the algorithm for hole-free graphs presented here is significantly improved since the algorithm given in [8] has a runtime of $O(n^4)$ and is not shown to be robust.

Small Graphs and Graph Classes

In a P_4, the edge with both endpoints having degree 2 is called *mid-edge*.

The *gem* is the graph consisting of a chordless path on four vertices with an additional universal vertex.

A *k-star* is a graph consisting of one vertex of degree k and k vertices of degree 1. A graph is called *star* if it is a k-star for any $k \geq 0$. The 0-star (a single vertex) and the 1-star (a single edge) are also called *trivial stars* and, consequently, all other stars are called *non-trivial stars*.

A *diamond* consist of two triangles that share exactly one edge. The shared edge is called the *mid-edge* of the diamond.

The graph *vase* equals the graph $P_2 + P_3$, that is, the disjoint union of an edge and a chordless path of length 2, with an additional universal vertex.

Two triangles that share exactly one vertex are called *butterfly*. The two edges that are not incident to the shared vertex are called the *outer edges* of the butterfly.

An (m, k)-*mouse* consists of two chordless cycles, one with length m and one with length k, that share exactly one vertex, say c, and an additional

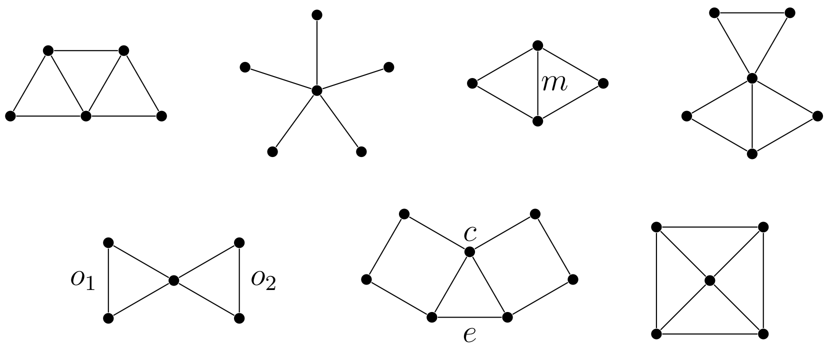

Figure 3.1: From left to right: The gem, the 5-star, the diamond with mid-edge m, the vase, the butterfly with outer edges o_1 and o_2, the 4-mouse with center vertex c and center edge e, and the 4-wheel.

edge c_1c_2, where c_1 is part of the one cycle and c_2 is part of the other cycle, such that $c{-}c_1{-}c_2{-}c$ is a triangle. The vertex c is called the *center vertex* of the (m, k)-mouse and the edge c_1c_2 is called *center edge*. A graph is called *k-mouse*, if it is a (k, k)-mouse and a graph is called *mouse*, if it is an (m, k)-mouse for any $m \geq 3$ and $k \geq 3$.

The *k-wheel* consists of a chordless cycle of length k and a universal vertex. Figure 3.1 depicts the gem, the 5-star, the diamond, the vase, the butterfly, the 4-mouse and the 4-wheel.

A graph is called *block graph*, if every block of the graph is a complete graph.

A *two-terminal graph* is a graph $G = (V, E)$ with two distinguished vertices $s, t \in V$, denoted by (G, s, t). For two disjoint two-terminal graphs (G_1, s_1, t_1) and (G_2, s_2, t_2), we define the following compositions:

- The *generalized series-composition* is either (G, s_1, t_1) or (G, s_1, t_2), where G is the union of G_1 and G_2 on $t_1 = s_2$.

- The *parallel-composition* is (G, s_1, t_1) where G is the union of G_1 and G_2 on $s_1 = s_2$ and $t_1 = t_2$.

A graph is called *generalized series-parallel two-terminal graph* if it is either $((\{s, t\}, \{st\}), s, t)$, that is, a simple edge with endpoints s and t, or the generalized series-composition or the parallel-composition of two generalized series-parallel two-terminal graphs. Consequently, a graph G is called *generalized series-parallel graph*, if for some pair $s, t \in V(G)$ it is a generalized series-parallel two-terminal graph (G, s, t).

Notions and Observations

The results in the following sections apply a simple observation about efficient edge dominating sets of triangles and induced cycles of length 4:

Observation 37. *Let $G = (V, E)$ be a graph. Every efficient edge dominating set D of G contains*

(i) exactly one edge of every triangle of G, and,

(ii) no edge of any induced C_4 of G.

Proof. Clearly, a triangle cannot contain two edges of an efficient edge set. Assume that no edge of a triangle, say $x–y–z–x$, is part of an efficient edge dominating set D. Since D dominates all edges, at least two vertices of x, y, and z must be in $V(D)$. Without loss of generality, let $\{x, y\} \subseteq V(D)$. Since $xy \notin D$, there are vertices x' and y' with $\{xx', yy'\} \subseteq D$. Hence, D is not efficient—this is a contradiction.

Assume that an induced C_4, say $a–b–c–d–a$, contains an edge of an efficient edge dominating set D, say $ab \in D$. Since D dominates cd, it is either $cd \in D$ or there is a vertex c' with $cc' \in D$ or a vertex d' with $dd' \in D$. In any case, D is not efficient—this is a contradiction. $\square$

One can easily check that Observation 37 implies that an efficiently edge dominatable graph cannot contain a K_4, a gem, a 4-wheel or a vase as induced subgraph:

Observation 38. *Let $G = (V, E)$ be a graph. If G is efficiently edge dominatable, then G is $\{K_4, gem, 4\text{-}wheel, vase\}$-free.*

For designing efficient algorithms, it is useful to detect edges that are in every efficient edge dominating set of a graph. The following observation lists some configurations that are easy to prove and/or simple implications from Observation 37:

Observation 39. *Let $G = (V, E)$ be a graph and $xy \in E$. If*

(i) xy is the mid-edge of an induced P_4, say $x'–x–y–y'$, and x' and y' have degree 1 in G, or,

(ii) xy is the mid-edge of an induced diamond, or,

(iii) xy is an outer edge of an induced butterfly, or,

(iv) xy is the center edge of an induced 4-mouse,

then every efficient edge dominating set of G contains xy.

This justifies the definition of mandatory edges:

Definition 4. *In a graph $G = (V, E)$, an edge xy is called*

- mandatory path edge*, if it meets the condition of Observation 39.i, and,*
- mandatory triangle edge*, if it meets the condition of Observation 39.ii, Observation 39.iii, or Observation 39.iv.*

An edge is called mandatory*, if it is a mandatory path edge or a mandatory triangle edge.*

In algorithms, we need an efficient way to check for a given vertex v, if v is part of one of the mentioned configurations for mandatory edges. The following two observations provide a tool for this task:

Observation 40. *Let $G = (V, E)$ be a graph. There is a vertex $v \in V$ such that*

(i)	$G[N(v)]$ *contains a triangle*	$\Leftrightarrow$	G *contains a* K_4
(ii)	$G[N(v)]$ *contains a* P_4	$\Leftrightarrow$	G *contains a gem*
(iii)	$G[N(v)]$ *contains a* C_4	$\Leftrightarrow$	G *contains a* 4*-wheel*
(iv)	$G[N(v)]$ *contains a* $P_2 + P_3$	$\Leftrightarrow$	G *contains a vase*
(v)	$G[N(v)]$ *contains a* P_3	$\Leftrightarrow$	G *contains a diamond*
(vi)	$G[N(v)]$ *contains a* $2K_2$	$\Leftrightarrow$	G *contains a butterfly*

as induced subgraphs.

Observation 41. *Let $G = (V, E)$ be an efficiently edge dominatable graph. The neighborhood $N(v)$ of every vertex $v \in V$ induces either*

- *the union of trivial stars or*
- *the union of a non-trivial star and an independent set*

in G.

Proof. Let $G = (V, E)$ be a graph that admits an efficient edge dominating set D and let $v \in V$. Assume that $G[N(v)]$ contains a P_4. Then, by Observation 40, G contains a gem—this is a contradiction to the existence of D by Observation 38. Analogously, an induced C_3 or C_4 in $G[N(v)]$

contradicts the K_4-freeness or 4-wheel-freeness of G respectively. Since every cycle of length 5 or more contains the P_4, $G[N(v)]$ is cycle-free.

Clearly, the connected components of $\{P_4, \text{cycle}\}$-free graphs are stars. Assume that there are two connected components F and H of $G[N(v)]$ that are non-trivial stars. Let $x{-}y{-}z$ be an induced P_3 of F and let $x'{-}y'{-}z'$ be an induced P_3 of H. Since $\{v, x, y, z\}$ and $\{v, x', y', z'\}$ clearly induce diamonds in G, the edges vx and vx' are mandatory. Hence, we have $\{vx, vx'\} \subseteq D$—this is a contradiction to the efficiency of D. $\square$

Throughout this chapter we assume that every graph $G = (V, E)$ is edge weighted with a weight function $\omega : E \to \mathbb{R}^+$. If a graph is not edge weighted, we simply assume $\omega(e) = 1$ for every $e \in E$.

3.1 Chordal bipartite graphs

In this section, we show:

Theorem 11. *Given a chordal bipartite graph,* MINIMUM WEIGHT EFFICIENT EDGE DOMINATION *is linear-time solvable.*

The core of the proof of Theorem 11 is a reduction from MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on chordal bipartite graphs to MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on K_4-free block graphs, a subclass of generalized series-parallel graphs. The idea of the reduction is mainly based on Observation 37.ii, since every non-trivial block of a chordal bipartite graph consists of one or more induced cycles of length 4. This means that no efficient edge dominating set contains an edge of a non-trivial block and, hence, the edges of the non-trivial blocks must be dominated by edges of trivial blocks. Due to the efficiency property, it turns out that the covered vertices of a non-trivial block belong to the same part of a bipartition of the graph:

Lemma 42. *Let $G = (V, E)$ be a graph and let B be a biconnected, chordal bipartite, induced subgraph of G of at least four vertices and let $V(B) = X \cup Y$ be a bipartition of B. If $D \subseteq E$ is an efficient edge dominating set of G, then either*

$$V(D) \cap V(B) = X$$

or

$$V(D) \cap V(B) = Y.$$

Proof. Since B contains at least four edges, $V(B) \cap V(D) \geq 2$.

If there is an $x \in V(D) \cap X$, let $y \in Y$ be any neighbor of x. By Observation 37.ii, we know that $xy \notin D$. If there is an edge $yy' \in D$ for some $y' \neq x$, then D is not efficient. Hence, $y \notin V(D)$. Let $x' \in X$ be any neighbor of y that is different from x, which must exist because B is biconnected. Since D must dominate the edge $x'y$, it must be $x' \in V(D)$. This argumentation can be repeated and, since B is biconnected, we get $V(D) \cup V(B) = X$.

If there is a $y \in V(D) \cap V(B) \cap Y$, an analogous argumentation gives $V(D) \cup V(B) = Y$. □

Since blocks are biconnected induced subgraphs, Lemma 42 implies: On chordal bipartite graphs, there are only two ways for every non-trivial block to cover its vertices by an efficient edge dominating set. We mimic this in the reduction by replacing the edges of the block by a gadget that is K_4-free and allows exactly two different efficient edge dominating sets.

Definition 5. *Let $G = (X \cup Y, E)$ be a chordal bipartite graph with edge weights ω and let $B_1, \ldots, B_k$ be its non-trivial blocks. The reduced graph $G' = (V', E')$ with edge weights ω' contains all vertices of G and all edges of trivial blocks of G. For every non-trivial block $B_i = (X_i \cup Y_i, E_B)$ of G, G' additionally contains*

- *the gadget G_i, which consists of the triangle $\{x_i, y_i, d_i\}$ and the vertex p_i that is adjacent only to d_i,*
- *the edge $x_i x$ for every $x \in X_i$, and,*
- *the edge $y_i y$ for every $y \in Y_i$.*

The weight function ω' is defined as:

- *$\omega'(e) := \omega(e)$ for all $e \in E \cap E'$, and,*
- *$\omega'(e) := 0$ for all $e \in E' \setminus E$.*

The reduction of Definition 5 is shown in Figure 3.2 for a single non-trivial block.

We show that the size of a minimum weighted efficient edge dominating set is invariant under this reduction:

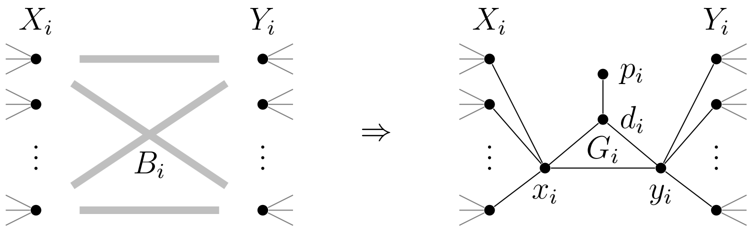

Figure 3.2: The non-trivial block B_i of G with bipartition $V(B_i) = X_i \cup Y_i$ (left) is substituted by the gadget G_i and additional edges in G' (right). The thick gray lines represent the edges of B_i.

Lemma 43. *Let $G = (V, E)$ be a chordal bipartite graph with edge weights ω and let $G' = (V', E')$ be the reduced graph with edge weights ω' as in Definition 5.*

The graph G admits an efficient edge dominating set of size Ω, if and only if G' admits an efficient edge dominating set of size Ω.

Proof. Let $B_1, \dots, B_k$ be the non-trivial blocks of G.

Let D be an efficient edge dominating set of G. We define D' as follows:

$$D' := D \cup D^T$$

where $D^T \subseteq \bigcup_{i=1}^{k} E(G_i)$ fulfills for all $1 \le i \le k$:

$$d_i y_i \in D^T \Leftrightarrow X_i \subseteq V(D) \text{ and } d_i x_i \in D^T \Leftrightarrow Y_i \subseteq V(D).$$

Since $D^T \subseteq E'$, $D \cap E(B_i) = \emptyset$ for all $1 \le i \le k$, by Observation 37.ii, and $D \subseteq E$, we get $D' \subseteq E'$. We show that D' is an efficient edge dominating set of G':

Notice that D' contains either $d_i x_i$ or $d_i y_i$ for every $i \in \{1, \dots, k\}$ by Lemma 42. Furthermore, by construction of G' and the definition of D^T, the edges in D^T have a minumum distance of 2 to each other and all edges in D. Since D is efficient in G and $G'[V]$ is an induced subgraph of G, this implies that D' is efficient.

Clearly, all edges in $E \cap E'$ are dominated by D and, hence, also by D'. Now consider the edges of G' that replace an arbitrary non-trivial block $B_i = (X_i \cup Y_i, E_i)$ of G. Assume without loss of generality that $d_i x_i \in D^T$. Then $x_i y_i$, $d_i p_i$, and $x x_i$ for all $x \in X_i$ are dominated by D'. By Lemma 42, $Y_i \subseteq D$ and, hence, the edges $y y_i$ for all $y \in Y_i$ are dominated by D' as well. This shows that D' is an efficient edge dominating set of G'.

Now let D' be an efficient edge dominating set of G'. We claim: For every $i \in \{1, \ldots, k\}$, D' contains either $x_i d_1$ or $y_i d_i$. Assume conversely that, without loss of generality, $d_1 x_1 \notin D'$ and $d_1 y_1 \notin D'$. By Observation 37.i, we have $x_1 y_1 \in D'$. Since D' is efficient, this implies $d_1 p_1 \notin D'$. But to dominate $d_1 p_1$, it must be $d_1 x_1 \in D'$, $d_1 y_1 \in D'$, or $d_1 p_1 \in D'$—this is a contradiction.

Let $D^T := (E' \setminus E) \cap D'$. We show that $D := D' \setminus D^T$ is an efficient edge dominating set of G:

As shown above, D^T contains either $x_i d_i$ or $y_i d_i$ for every $i \in \{1, \ldots, k\}$. Since D' is efficient, D^T contains exactly these edges.

For every $i \in \{1, \ldots, k\}$, either the edges between X_i and x or the edges between Y_i and y are not dominated by D^T. Since D' dominates all edges of G', these edges are dominated by D, which implies that either $X_i \subseteq V(D)$ or $Y_i \subseteq V(D)$. Every edge of a non-trivial block B_i of G is incident to a vertex of X_i and a vertex of Y_i, what means that it is dominated by D. Since D clearly dominates all edges in $E \cap E'$, it is an efficient edge dominating set of G.

Since in both cases $\omega'(D^T) = 0$ and $\omega(D) = \omega'(D)$, it follows $\omega(D) = \omega'(D')$. □

The reduction of Definition 5 produces a graph whose non-trivial blocks are triangles because all edges of non-trivial blocks are removed and a triangle is added. This means in particular that the resulting graph a K_4-free block graph.

One can easily check that K_4-free block graphs are contained in the class of generalized series-parallel graphs. For these graphs, there is an algorithm that solves MINIMUM WEIGHT EFFICIENT EDGE DOMINATION in linear time given by Lu, Ko and Tang in [70].

This permits the algorithm shown in Table 3.1. It applies the reduction of Definition 5 and solves MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on the resulting graph in linear time.

Lemma 44. *Algorithm* `MWEED-ChordalBipartite` *is correct and runs in time $O(n + m)$.*

Proof. Since the algorithm obviously constructs G' according to Definition 5 before Step 6, the correctness follows from Lemma 43.

A bipartition of a graph can be found in linear time by using DFS in Step 2.

In Step 3, the blocks of a graph can be determined in linear time using algorithms for testing 2-vertex connectivity, for example the algorithm of

Algorithm: `MWEED-ChordalBipartite`
Input: A chordal bipartite graph $G = (V, E)$ with edge weights ω.
Output: A minimum efficient edge dominating set D, if one exists.

1. Set $G' = (V', E') := (V, E)$ and $\omega' := \omega$.
2. Determine a bipartition $V = X \cup Y$ of G. */* by DFS */*
3. Determine the non-trivial blocks $B_1, \ldots, B_k$ of G. */* by Hopcroft-Tarjan */*
4. For every $e \in E$: If e is part of some non-trivial block, set $E' := E' \setminus \{e\}$.
5. For every non-trivial block B_i:
   1. Set $V' := V' \cup \{x_i, y_i, d_i, p_i\}$ and $E' := E' \cup \{d_i p_i, d_i x_i, d_i y_i, x_i y_i\}$ and $\omega'(d_i p_i) = \omega'(d_i x_i) = \omega'(d_i y_i) = \omega'(x_i y_i) := 0$.
   2. For every $x \in V(B_i) \cap X$: Set $E' := E' \cup \{x x_i\}$ and $\omega'(x x_i) := 0$.
   3. For every $y \in V(B_i) \cap Y$: Set $E' := E' \cup \{y y_i\}$ and $\omega'(y y_i) := 0$.

— *G' is a K_4-free block graph now.* —

6. Determine a minimum weighted efficient edge dominating set D' of G'. If it does not exist, `STOP:` G `is not efficient edge dominatable`.
7. `Return` $D := D' \cap E$.

Table 3.1: A linear-time algorithm for MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on chordal bipartite graphs.

Hopcroft and Tarjan [56] or a very simple algorithm recently given by Schmidt [87].

When determining the blocks of G, we can label every edge that is part of a non-trivial block and every vertex with the non-trivial blocks it is contained in. The labels of the edges can be used to remove the edges of non-trivial blocks in linear time in Step 4. The labels of the vertices can be used to implement Steps 5.2 and 5.3 in accumulated linear time because Fact 1 guarantees that at most $2|V|$ labels are given to the vertices. Clearly, Step 5.1 can be done in constant time and, since $k \leq n$, this accumulates to linear time.

Since the construction of G' takes only linear time, also its size is linear in the size of G. Hence, using the algorithm for MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on generalized series-parallel graphs given in [70], Step 6 can be done in linear time. □

This completes the proof of Theorem 11.

Unfortunately, the algorithm in the proof of Theorem 11 requires that the input graph is chordal bipartite. If this is not already known, the runtime of a robust algorithm depends on the complexity of the recognition of chordal bipartite graphs. With the best known recognition algorithms, derived from [74, 83, 88], we get:

Corollary 45. MINIMUM WEIGHT EFFICIENT EDGE DOMINATION *is solvable in time $O(\min(m \log n, n^2))$ on chordal bipartite graphs.*

3.2 Hole-free graphs

This section shows:

Theorem 12. MINIMUM WEIGHT EFFICIENT EDGE DOMINATION *is solvable in time $O(nm)$ on hole-free graphs in a robust way.*

The proof is similar to the technique used in Section 3.1, as we reduce the problem to MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on K_4-free block graphs. Since the blocks of hole-free graphs are more complex than chordal bipartite blocks, we first need some additional preparations.

Let $G = (V, E)$ be a graph. As we want to design a robust algorithm, we have to cover the case that G is not hole-free.

Our algorithm starts with checking whether G fulfills the necessary condition of Observation 38, that is, it checks whether G is $\{K_4, \text{gem}, 4\text{-wheel}, \text{vase}\}$-free. A way to do this check efficiently is given by Observation 40 and its implication, Observation 41. It states that the neighborhood of every vertex induces either a union of trivial stars or a union of a non-trivial star and an independent set, if G is efficiently edge dominatable. If a check for this condition fails, then G admits no efficient edge dominating set. Hence, from now on we can assume that G is $\{K_4, \mathrm{gem}, 4\text{-wheel}, \mathrm{vase}\}$-free.

We first reduce the problem to the case, where the input graph has no mandatory triangle edges. This modification preserves the efficient edge dominating sets of G and their weights. The idea is to replace mandatory triangle edges by mandatory path edges. Since the replacement works for every mandatory edge, we can define it in general:

Definition 6. *Let $G = (V, E)$ be a graph with edge weights ω and let $xy \in E$ be a mandatory edge. We define $G_{xy} = (V_{xy}, E_{xy})$ with*

$$V_{xy} := V \cup \{x', y'\} \text{ for new vertices } x' \text{ and } y' \text{ and}$$
$$E_{xy} := \{e \in E \mid e \cap \{x, y\} = \emptyset\} \cup \{xz \mid xz \in E \text{ or } yz \in E\} \cup \{xx', yy'\},$$

and ω_{xy} with

$$\omega_{xy}(e) := \omega(e) \text{ for all } e \in E_{xy} \cap E \text{ and}$$
$$\omega_{xy}(e) := 0 \text{ for all } e \in E_{xy} \setminus E,$$

that means, in G_{xy}, all edges of G that are incident to y are switched from y to x and there are two new pending vertices x' and y' such that $x'xyy'$ is an induced P_4 in G_{xy}.

This modification preserves all efficient edge dominating sets:

Lemma 46. *Let $G = (V, E)$ be a graph with edge weights ω, let $D \subseteq V$, let $xy \in E$ be a mandatory edge and let G_{xy} and ω_{xy} be defined as in Definition 6. Then*

$$D \text{ is an efficient edge dominating set of } G \text{ of size } \Omega,$$

if and only if

$$D \text{ is an efficient edge dominating set of } G_{xy} \text{ of size } \Omega.$$

Proof. Let D be an efficient edge dominating set of G. Since xy is mandatory, $xy \in D$. Hence, we have $N_G(xy) \cap V(D) = \emptyset$ because otherwise D would not be efficient. With this, D is efficient and edge dominating in G_{xy} because D

is efficient and edge dominating in $G - N_G[xy] = G_{xy} - N_{G_{xy}}[xy]$ and every edge that is adjacent to xy in G_{xy}, in particular xx' and yy', is dominated by xy.

Conversely, let D_{xy} be an efficient edge dominating set of G_{xy}. Since $x'–x–y–y'–x'$ is an induced P_4 in G_{xy} and x' and y' are of degree 1, xy is a mandatory path edge and we have $xy \in D_{xy}$. Hence, $N_{G_{xy}}(xy) \cap V(D_{xy}) = \emptyset$ because otherwise D_{xy} would not be efficient. With this, D_{xy} is efficient and edge dominating in G because D_{xy} is efficient and edge dominating in $G_{xy} - N_{G_{xy}}[xy] = G - N_G[xy]$ and every edge that is adjacent to xy in G is dominated by xy. $\square$

Since we want to apply this modification to hole-free graphs, we have to show that hole-free graphs are closed under this modification.

Lemma 47. *Let $G = (V, E)$ be a graph, let $xy \in E$ be a mandatory edge of G and let G_{xy} and ω_{xy} be defined as in Definition 6. For every induced cycle C_{xy} of G_{xy} of length k, there is an induced cycle C of G of length at least k. Hence, if G is $\{C_k, C_{k+1}, \dots\}$-free, then G_{xy} is $\{C_k, C_{k+1}, \dots\}$-free for any k.*

Proof. Let C_{xy} be an induced cycle of G_{xy} on vertices $V_C \subseteq V_{xy}$. Clearly, $\{x', y', y\} \cap V_C = \emptyset$. If $x \notin V_C$, C_{xy} is also an induced cycle in G. Hence, let $x \in V_C$. Let $a \in V_C$ and $b \in V_C$ be the neighbors of x in C_{xy}. If $bx \in E$, then C_{xy} is an induced cycle in G. Hence, let $bx \notin E$. If $ay \in E$, then $(V_C \setminus \{x\}) \cup \{y\}$ induces a cycle in G with the same length as in G_{xy}. If $ay \notin E$, then $V_C \cup \{y\}$ induces a cycle in G that is longer than C_{xy}. $\square$

The use of this modification is to simplify the structure of the blocks by decreasing the number of triangles. The following two observations together imply that constructing G_{xy} for an efficiently edge dominatable graph G destroys every triangle which contains the edge xy and introduces no new triangles.

Observation 48. *Let $G = (V, E)$ be a graph and let $xy \in E$ be a mandatory edge of G. If G has an efficient edge dominating set, then $N(xy)$ is independent.*

Proof. Assume that $N(xy)$ contains an edge, say ab. If $\{ax, bx\} \subseteq E$ or $\{ay, by\} \subseteq E$, then xy is incident to the triangle $x–a–b–x$ or the triangle $y–a–b–y$. Since xy is mandatory, then G is not efficiently edge dominatable by Observation 37.i—this is a contradiction. If this is not the case, then we can assume without loss of generality that $\{ax, by\} \subseteq E$ and $\{bx, ay\} \cap E = \emptyset$. But then xy is part of the induced cycle $x–y–a–b–x$, and, hence, G is not efficiently edge dominatable by Observation 37.ii—this is a contradiction. $\square$

Observation 49. *Let $G = (V, E)$ be a graph, let $xy \in E$ be a mandatory triangle edge of G and let G_{xy} be defined as in Definition 6. Let T be the set of all triangles of G and $T' \subseteq T$ the set of all triangles that contain the edge xy.*

If $N(xy)$ is independent in G, then the set T_{xy} of triangles of G_{xy} is $T_{xy} = T \setminus T'$.

Proof. Every triangle of the form $x–y–z$ of G does not exist in G_{xy} because the edge yz is removed. Since $G[N(xy)]$ contains no edges, replacing an edge yz for some $z \in V \setminus \{x, y\}$ by the edge xz cannot introduce a new triangle. $\square$

Our goal is to repeat the modification until there are no more mandatory triangle edges. Hence, we have to identify mandatory triangle edges efficiently.

By Observation 40, the mid-edges of induced diamonds and the outer edges of induced butterflies can easily be found by considering the neighborhood of every vertex and check it for an induced P_3 or an induced $2K_2$. Consequently, we identify all edges xy that are mid-edges of induced diamonds or outer edges of induced butterflies and check if the neighborhood of xy is independent. If so, we construct G_{xy}, otherwise, we have a proof that G admits no efficient edge dominating set and, thus, we stop. At the end, the resulting graph G is $\{\text{diamond}, \text{butterfly}\}$-free, by Observation 49. Notice that, since G is also K_4-free as assumed above, every vertex is in at most one triangle.

Since G is $\{\text{diamond}, \text{butterfly}\}$-free, all remaining mandatory triangle edges are center edges of induced 4-mice. To detect these, we search for all induced mice in G. Since the triangles of G are pairwise disjoint, there is no induced $(3, k)$-mouse or $(k, 3)$-mouse for any $k \geq 3$ in G. Moreover, as the number of triangles is at most $|V|$, the number of mice is at most $|V|$. Hence, every vertex of G can be the center vertex of at most one induced mouse. This leads to the following algorithm: For every vertex v, check if $G[N(v)]$ contains an edge, that is, check if v is part of a triangle. Notice that there can be at most one edge in $G[N(v)]$ because G is $\{\text{diamond}, \text{butterfly}\}$-free. If such an edge exists, say xy, then let P_{vx} (respectively P_{vy}) be a shortest path from v to x (respectively from v to y) that neither uses the edge vx (respectively vy) nor the vertex y (respectively x), if such a path exists. These paths can easily be found by breadth-first-search. If both paths exist, then G contains an induced (m, k)-mouse, where m is the length of P_{vx} and k is the length of P_{vy}. If $m > 4$ or $k > 4$, then $P_{vx} \cup \{vx\}$ or $P_{vy} \cup \{vy\}$ induce a hole in G, which proves that G is not hole-free. If $m = k = 4$, then xy is a mandatory triangle edge and, again, we check if its neighborhood is independent and if so, we apply the modification of Definition 6. After checking this for every vertex, by Observation 49, the graph is $\{\text{diamond}, \text{butterfly}, \text{mouse}\}$-free or we have a proof that G contains an induced hole or admits no efficient edge dominating set.

The procedure for reducing the mandatory triangle edges is shown in Table 3.2. It is a straightforward implementation of the foregoing considerations. We show:

Lemma 50. *Procedure* `ReduceMandatoryTriangleEdges-Holefree` *is correct and runs in time $O(|V| \cdot |E|)$. The output graph G' has at most $5|V|$ vertices and at most $5|E|$ edges. If G is hole-free, then G' is hole-free.*

Proof. Assume that G is given as one incidence list per vertex.

The loop of Step 2 is executed $|V|$ times. The neighborhood of a vertex is given by its incidence list. To achieve a constant time lookup later, we label every vertex of $N(v)$ in Step 2.1, which takes at most time $O(|V|)$. We can count the degree of all vertices of $N(v)$ in $G[N(v)]$ by initializing the degree of every vertex with 0, then considering every edge xy of G and increase the degree of x and y by one, if both, x and y, are labeled. Clearly, the maximum and the sum can be computed alongside. Hence, Step 2.2 can be done in time $O(|E|)$.

The correctness of Step 2.3 follows from Observation 38 and Observation 40 and the test clearly can be done in constant time. The correctness of Steps 2.4 and 2.5 follows from Observation 39 and Observation 40. Step 2.4 can be done in constant time and Step 2.5 takes at most time $O(|E|)$. Hence, the execution of the loop in Step 2 takes at most time $O(|V| \cdot |E|)$. When the loop finishes, M contains the mid-edge of all diamonds and the outer edges of all butterflies of G.

Step 3 is correct because if $|M| > |V|$, there is a vertex that is incident to two mandatory edges—this is a contradiction to the existence of an efficient edge dominating set.

By Step 3, the loop in Step 4 is executed at most $|V|$ times. The neighborhood $N(xy)$ of an edge xy can be constructed in time $O(|N(xy)|) \subseteq O(|V|)$ using the incidence lists of x and y. Again, we label the vertices of $N(xy)$. Then, we can check if $N(xy)$ is independent in G' by considering every edge vw of G' and check if at most one of v and w is labeled. This takes time $O(|E'|)$. The correctness of Step 4.1 directly follows from Observation 48.

Procedure: `ReduceMandatoryTriangleEdges-Holefree`
Input: A connected graph $G = (V, E)$ with edge weights ω.
Output: A connected $\{K_4, \text{gem}, \text{diamond}, \text{butterfly}, \text{mouse}\}$-free graph G' with edge weights ω' such that G and G' have the same efficient edge dominating sets or the statement that G is not hole-free or the statement that G is not efficiently edge dominatable.

1. Set $G' = (V', E') := (V, E)$ and $M := \emptyset$.
2. For every $v \in V$:
   1. Determine $N(v)$.
   2. Count $\deg_{G[N(v)]}(w)$ for every $w \in N(v)$ and calculate the sum s and the maximum m of all these values.
   3. If $m > 1$ and $m < s - m$, `STOP:` G `is not efficient edge dominatable`. /* *found K_4, gem, $4-$wheel or vase in G'* */
   4. If $m > 1$, set $M := M \cup \{vw\}$, where $w \in N(v)$ with $m = \deg_{G[N(v)]}(w)$. /* *mid-edge of a diamonds* */
   5. If $m = 1$, for every edge $xy \in G[N(v)]$: Set $M := M \cup \{xy\}$. /* *outer edges of butterflies* */
3. If $|M| \geq |V|$, `STOP:` G `is not efficient edge dominatable`.
4. For every $xy \in M$:
   1. Check if $N(xy)$ is an independent set in G'. If not: `STOP:` G `is not efficient edge dominatable`. /* *Observation 48* */
   2. Set $G' := G'_{xy}$ and $\omega' := \omega'_{xy}$ according to Definition 6.

— *G' is $\{K_4$, gem, diamond, butterfly$\}$-free now.* —

5. For every $v \in V$ such that $G'[N(v)]$ contains an edge xy:
   1. Find a shortest path P_{vx} from v to x in $(G' - y) - vx$ and a shortest path P_{vy} from v to y in $(G' - x) - vy$.
   2. If one of these paths exists and has length greater 3, `STOP:` G `is not hole-free`.
   3. If both paths exist, set $G' := G'_{xy}$ and $\omega' := \omega'_{xy}$. /* *4-mouse* */
6. `Return` G'.

Table 3.2: The procedure `ReduceMandatoryTriangleEdges-Holefree`.

Step 4.2 preserves the efficient edge dominating sets of G' by Lemma 46. The modification can be done in time $O(|V'|)$ because the addition of a constant number of vertices and edges takes constant time using incidence lists and switching all edges from y to x can be done by considering every neighbor of y, hence in time $O(|V'|)$. Hence, the loop in Step 4 takes at most time $O(|V| \cdot |V'|)$.

Observation 49 guarantees that G' is $\{\text{diamond}, \text{butterfly}\}$-free when the loop finishes because all triangles of diamonds and butterflies are reduced without introducing new triangles. After each iteration of the loop, G' contains 2 vertices and 2 edges more than before. Since the loop is executed at most $|V|$ times, when it finishes, we have $|V'| \leq 3|V|$ and $|E'| \leq 3|E|$.

The loop in Step 5 is executed at most $|V|$ times. To test if $G'[N(v)]$ contains an edge takes at most time $O(|E'|)$ because we can label all vertices of $N(v)$ like above and check for every edge of G if it has both endpoints in $N(v)$. Step 5.1 can be done by starting two breadth-first-searches, while one search omits the vertex y and the edge vx and the other search omits the vertex x and the edge vy. Since G' is connected, breadth-first-search takes at most time $O(|E'|)$. Step 5.2 is correct because if a path, say P_{vx}, has length 4 or greater, then the vertices of P_{vx} induce a hole in G. If both paths exist, they have length 4 because otherwise there is a diamond or butterfly in G' or we exited in Step 5.2. Hence, there is a 4-mouse in G' and xy is a mandatory triangle edge by Observation 39.iv. Thus, Step 5.3 preserves the efficient edge dominating sets by Lemma 46. It takes at most time $O(|V'|)$ as Step 4.2.

Observation 49 guarantees that G' is mouse-free when the loop finishes because all triangles of mice are reduced without introducing new triangles. To get rid of all triangles, notice that it is sufficient to regard the vertices of V only because all vertices in $V' \setminus V$ are in no triangle of G' by construction. After each iteration of the loop, G' contains at most 2 vertices and 2 edges more than before. Since the loop is executed at most $|V|$ times and, before the loop starts, we have $|V'| \leq 3|V|$ and $|E'| \leq 3|E|$, when it finishes we have $|V'| \leq 5|V|$ and $|E'| \leq 5|E|$. Since the size of G' never decreases during the execution of the procedure, this can be used as a bound for $|V'|$ and $|E'|$ in every step of the procedure. Hence, the overall running time is $O(|V| \cdot \max(|V'|, |E'|)) = O(|V| \cdot \max(|V|, |E|))$, which equals $O(|V| \cdot |E|)$ on connected graphs.

Lemma 47 assures in Steps 4.2 and 5.3 that G' is hole-free, if G was hole-free. □

Now we are ready to perform the reduction to K_4-free block graphs.

Hence, we consider the blocks of G. A block with three vertices clearly is a triangle. Every block with at least four vertices contains an induced C_4 because otherwise there is an induced K_4 or diamond in G. Our goal is to reduce the blocks of G to triangles. Consequently, the blocks of at least four vertices are of interest. Let B be a block of G with at least four vertices. Since G is $\{\text{diamond}, \text{butterfly}, \text{mouse}\}$-free, the triangles of B have the following useful property:

Fact 51. *Every triangle of B has exactly one vertex with degree 2 in B.*

This allows the following definition:

Definition 7. *Let $T \subset V(B)$ be the set of all vertices of B that are part of a triangle of B. The vertices $T^t := \{t \in T \mid \deg_B(t) = 2\}$ are called* thorns *of B.*

We call $B' := B - T^t$ the thornless block *of B.*

Clearly, by Fact 51, the thornless blocks of G are triangle-free and, hence, G is hole-free, if and only if all thornless blocks are chordal bipartite. Thus, our algorithm determines T^t for every block B of at least four vertices and checks the corresponding thornless block B' for being chordal bipartite. If this is not the case for at least one block, we have a proof that G is not hole-free.

Hence, from now on we can assume that G is hole-free and, consequently, that all thornless blocks are chordal bipartite. This enables the following definition:

Definition 8. *Let $V(B') = X \cup Y$ be a bipartition of B'. We define*

$$X^{\triangle} := X \cap T \text{ and } Y^{\triangle} := Y \cap T,$$

that is, a bipartition of the vertices of B' that are in a triangle of B. In contrast, we define

$$X^{\not\triangle} := X \setminus T \text{ and } Y^{\not\triangle} := Y \setminus T,$$

that is, a bipartition of the vertices of B' that are not in a triangle of B. Consequently, we define

$$N_X^{\triangle} := N(X^{\triangle}) \setminus V(B) \text{ and } N_Y^{\triangle} := N(Y^{\triangle}) \setminus V(B)$$

and

$$N_X^{\not\triangle} := N(X^{\not\triangle}) \setminus V(B) \text{ and } N_Y^{\not\triangle} := N(Y^{\not\triangle}) \setminus V(B)$$

as well as

$$N^t := N(T^t) \setminus V(B),$$

that is, the neighbors of $X^{\triangle}$, $Y^{\triangle}$, $X^{\not\triangle}$, $Y^{\not\triangle}$ and T^t outside of B.

Furthermore, the thornless blocks of G are biconnected, chordal bipartite subgraphs of G, so we can apply Lemma 42. That is, every efficient edge dominating set D of G covers either X or Y of every thornless block of G. By Fact 51, every triangle of B has exactly one edge in B'. Hence, Observation 37.i and Lemma 42 imply that every efficient edge dominating set of G contains one of the two edges of every triangle that are not in B'. We summarize this in the following corollary:

Corollary 52. *For every efficient edge dominating set D of G, either*

$$V(D) \cap V(B) = X \cup T^t \text{ or}$$
$$V(D) \cap V(B) = Y \cup T^t.$$

This implies that $E(X, Y) \cap D = \emptyset$ and that either

$$E(T^t, X^\triangle) \subset D \text{ and } E(T^t, Y^\triangle) \cap D = \emptyset \text{ or}$$
$$E(T^t, Y^\triangle) \subset D \text{ and } E(T^t, X^\triangle) \cap D = \emptyset,$$

and, hence,

$$E(T^t, N^t) \cap D = \emptyset,$$
$$E(X^\triangle, N_X^\triangle) \cap D = \emptyset, \text{ and}$$
$$E(Y^\triangle, N_Y^\triangle) \cap D = \emptyset.$$

We develop an extended version of the reduction used in Section 3.1.

Definition 9. *The reduced graph $G_B = (V_B, E_B)$ with edge weights ω_B results from G and ω by removing the vertices of T and the edges of $E(B)$ and adding the gadget H that contains*

- *the new triangle $\{x, y, d\}$ and the new vertex p that is adjacent only to d,*
- *the edge xv for every $v \in X^{\not\triangle}$ and every $v \in N_Y^\triangle$,*
- *the edge yv for every $v \in Y^{\not\triangle}$ and every $v \in N_X^\triangle$, and,*
- *the edge dv for every $v \in N^t$.*

and by defining ω_B as

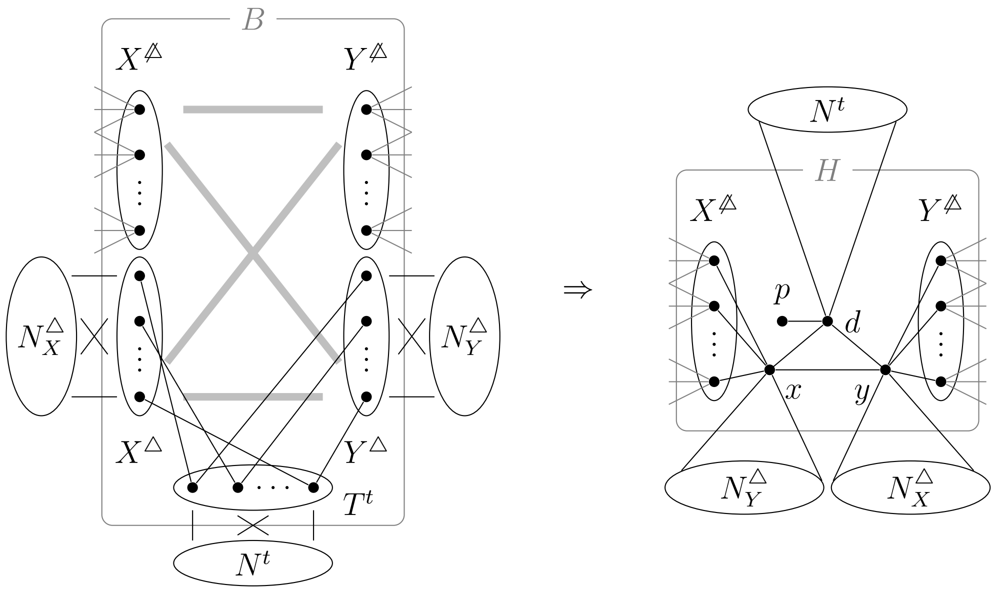

Figure 3.3: The block B of G (left) is substituted by the gadget H and additional edges in G_B (right). The thick gray lines represent the edges of B'. Clearly, every vertex of $X^{\triangle}$ has exactly one neighbor in $Y^{\triangle}$ and vice versa, but these edges are also represented by the thick gray lines, so they are not printed explicitly.

- $\omega_B(e) := \omega(e)$ *for all* $e \in E \cap E_B$,
- $\omega_B(dx) := \omega\left(E(Y^{\triangle}, N_Y^{\triangle})\right)$,
- $\omega_B(dy) := \omega\left(E(X^{\triangle}, N_X^{\triangle})\right)$, *and,*
- $\omega_B(e) := 0$ *for all* $e \in E_B \setminus E$ *except* $e = dx$ *and* $e = dy$.

The reduction described in Definition 9 is shown in Figure 3.3. We show that it preserves the sizes of all efficient edge dominating sets of G:

Lemma 53. *Let* $G = (V, E)$ *be a* $\{$*diamond, butterfly, mouse, hole*$\}$*-free graph with edge weights* ω*, let* B *be a block of* G *with at least four vertices and let* $G_B = (V_B, E_B)$ *be the reduced graph with edge weights* ω_B *as in Definition 9.*

There is an efficient edge dominating set D *of* G *with* $\omega(D) = \Omega$*, if and only if there is an efficient edge dominating set* D_B *of* G_B *with* $\omega_B(D_B) = \Omega$.

Proof. For the first direction, let D be an efficient edge dominating set of G. We construct D_B as follows:

$$D_B := (D \setminus E(B)) \cup \{dy\}\,,\ \text{if } V(D) \cap V(B) = X \cup T^t$$

and

$$D_B := (D \setminus E(B)) \cup \{dx\}\,,\ \text{if } V(D) \cap V(B) = Y \cup T^t.$$

We know from Corollary 52 that either the first or the second condition holds, so D_B is well defined. Corollary 52 also states that $E(T, V \setminus T) \cap D = \emptyset$. Since G_B contains all edges of G except $E(B)$ and the edges incident to T, we clearly have $D_B \subseteq E_B$. Without loss of generality assume that $V(D) \cap V(B) = Y \cup T^t$.

Assume that D_B is not efficient. Since D is efficient and G_B introduces no edges between vertices of V, there must be an edge in $D \setminus E(B)$ with distance 1 to dx. Hence, there is a vertex $v \in V(D \setminus E(B))$ with $v \in X^{\not\triangle}$, $v \in N_Y^{\triangle}$ or $v \in N^t$. Since $V(D) \cap V(B) = Y \cup T^t$, it cannot be $v \in X^{\not\triangle}$. Corollary 52 implies that all edges of $E(T^t, Y^{\triangle})$ are in D. Since D is efficient, this means that $v \notin N^{\triangle}$ and $v \notin N^t$. Hence, v cannot exist—this is a contradiction to the assumption that D_B is not efficient.

Since we have $dx \in D_B$, the edges dp, dx, dy, xy and all edges in $E(d, N^t)$, $E(x, X^{\not\triangle})$, and $E(x, N_Y^{\triangle})$ are dominated by D_B.

By Corollary 52, all vertices of $Y^{\not\triangle}$ are covered by edges that are outside of B. These edges also exist in D_B and, hence, all edges that have at least one endpoint in $Y^{\not\triangle}$ are dominated by D_B.

Again by Corollary 52, $X^{\triangle} \cap D = \emptyset$, which implies that D covers every vertex of $N_X^{\triangle}$ and every vertex of $N(X^{\not\triangle}) \setminus Y$ with an edge that is outside of B. These edges also exist in D_B and, hence, all edges that have at least one endpoint in $N_X^{\triangle}$ or $N(X^{\not\triangle}) \setminus Y$ are dominated by D_B.

Finally, every edge of G_B that has no endpoint in $V(H)$, N^t, $N_X^{\triangle}$ or $N_Y^{\triangle}$ is also an edge in G and dominated by $D \cap D_B$. This shows that D_B is edge dominating in G_B.

For the other direction, let D_B be an efficient edge dominating set of G_B. We construct D as follows:

$$D := (D_B \setminus \{dx\}) \cup E(T^t, Y^{\triangle}),\ \text{if } dx \in D_B$$

and

$$D := (D_B \setminus \{dy\}) \cup E(T^t, X^{\triangle}),\ \text{if } dy \in D_B.$$

Every efficient edge dominating set of G_B contains either dx or dy because otherwise dp is not dominated or no edge of the triangle d, x, y is chosen, a contradiction to Observation 37.i. Hence, D is well defined. Furthermore, this implies that D_B does not contain dp or xy, no edge of $E(x, N_Y^{\triangle})$ or $E(y, N_X^{\triangle})$, and no edge that has at least one endpoint in N^t. Hence, $D \subseteq E$. Without loss of generality assume that $dx \in D_B$.

Assume that D is not efficient. Clearly, $D \cap D_B$ is efficient. Since X, Y and T^t are independent sets in G, the edges of $E(T^t, Y^{\triangle})$ are efficient. Hence, there is an edge in $D \cap D_B$ that has distance 0 or 1 to an edge of $E(T^t, Y^{\triangle})$. Since $N^t \cap V(D_B) = \emptyset$ and the edges $E(Y^{\triangle}, N_Y^{\triangle})$, as well as the edges $E(X, Y)$, do not exist in G_B, this edge must be of the form vw with $v \in N_Y^{\triangle}$ or $v \in X^{\not\triangle}$ and $w \in V \setminus E(B)$. This edge is also in D_B and has distance 1 to dx—this is a contradiction to the efficiency of D_B. Hence, D is efficient.

Clearly, all edges of $G[T]$ are dominated by D, as well as all edges of $E(Y^{\triangle}, N_Y^{\triangle})$ and $E(Y^{\triangle}, X)$.

Since $dx \in D_B$ and $E(y, Y^{\not\triangle}) \cap D_B = \emptyset$, to dominate the edges in $E(y, Y^{\not\triangle})$, there are $\ell = |Y^{\not\triangle}|$ edges $\{e_1, \dots, e_\ell\} \subseteq D_B$, each having exactly one endpoint in $Y^{\not\triangle}$. By definition, these edges are also in D and, hence, D also dominates the edges in $E(Y^{\not\triangle}, X)$ in G. Analogous argumentations work for the edges in $E(y, N_X^{\triangle})$ and $E(X^{\not\triangle}, V_B \setminus \{x\})$ and we get that D dominates the edges in $E(X^{\triangle}, N_X^{\triangle})$ and $E(X^{\not\triangle}, V \setminus Y)$.

Finally, every edge of G that has no endpoint in $V(B)$, N^t, $N_X^{\triangle}$ or $N_Y^{\triangle}$ is also an edge in G_B and dominated by $D \cap D_B$. This shows that D is edge dominating in G.

It is easy to check that $\omega(D) = \omega_B(D_B)$, if D and D_B are constructed as described above. This completes the proof. $\square$

The reduction replaces the edges and some vertices of a block B of at least four vertices by a triangle with a pending edge. Since B is a maximal biconnected subgraph of G, all vertices of G that have a neighbor in B are pairwise unconnected in $G - E(B)$. Hence, the reduction does not introduce any kind of cycle, in particular it introduces no diamonds, butterflies, mice or holes. Moreover, it reduces the number of blocks of at least four vertices. This means, if we repeat the reduction until the resulting graph G' does not contain any block of at least four vertices, we have a K_4-free block graph G' that admits an efficient edge dominating set of size Ω, if and only if G admits an efficient edge dominating set of size Ω. This enables us to give a robust MINIMUM WEIGHT EFFICIENT EDGE DOMINATION algorithm for hole-free graphs shown in Table 3.3.

Algorithm: `MWEED-Holefree`
Input: A connected graph $G = (V, E)$ with edge weights ω.
Output: A minimum weighted efficient edge dominating set D, if one exists, or a proof that G is not hole-free or not efficiently edge dominatable.

1. Set $G :=$ `ReduceMandatoryTriangleEdges-Holefree`(G).

— *G' is $\{K_4$, gem, diamond, butterfly, mouse$\}$-free.* —

2. Set $G' = (V', E') := G$ and $\omega' := \omega$.
3. Determine the blocks of G.
4. For every block B of G with at least four vertices:
   1. Count $\deg_B(v)$ for every $v \in V(B)$.
   2. Set $T^t := \{v \mid v \in V(B), \deg_B(v) = 2, \text{ and } N_B(v) \in E'\}$.
   3. Check if $B - T^t$ is chordal bipartite. If not: `STOP:` G `is not hole-free`.
   4. Find a bipartition $V(B - T^t) = X \cup Y$.
   5. Set $G' := G'_B$ and $\omega' := \omega'_B$ according to Definition 9.

— *G' is a K_4-free block graph now.* —

5. Determine a minimum weighted efficient edge dominating set D' of G'. If it does not exist, `STOP:` G `is not efficient edge dominatable`.
6. For every block B of G with at least four vertices:
   1. If $dx \in D'$, set $D' := D' \setminus \{dx\} \cup E(T^t, Y)$.
   2. If $dy \in D'$, set $D' := D' \setminus \{dy\} \cup E(T^t, X)$.
7. `Return` D'.

Table 3.3: A robust algorithm for MINIMUM WEIGHT EFFICIENT EDGE DOMINATION on hole-free graphs.

Lemma 54. *Algorithm* `MWEED-Holefree` *is correct and runs in time* $O(|V| \cdot |E|)$.

Proof. Assume that G is given as one incidence list per vertex.

By Lemma 50, Step 1 runs in time $O(|V| \cdot |E|)$. Furthermore, after its execution, G is $\{K_4, \text{gem}, \text{diamond}, \text{butterfly}, \text{mouse}\}$-free, G is hole-free if the input graph was hole-free, G has the same efficient edge dominating sets as the input graph, and the size was increased only by a constant factor. Hence, it is correct to continue on the resulting graph G instead of the input graph.

Since G is connected, Step 3 can be done in linear time using algorithms for testing 2-vertex-connectivity, for example the well-known algorithm of Hopcroft and Tarjan [56] or a very simple algorithm recently given by Schmidt [87]. We can assume that we get a list of vertices and a list of edges for every block.

Step 4.1 can be done in time $O(|E(B)|)$ by initializing the degree of every vertex of B with 0, considering every edge $xy \in E(B)$ and increase the degree of x and y. Step 4.2 can be done in time $O(|V(B)|)$ be checking for every vertex of B with degree 2 in B, if its neighborhood is an edge. The induced subgraph $B - T^t$ can be constructed in time $O(|V(B)| + |E(B)|)$ by labeling every vertex of T^t and creating a copy of B while omitting every labeled vertex and every edge with at least one labeled endpoint. Step 4.3 can be done in time $O(\min(|E(B)| \log |V(B)|, |V(B)|^2))$ by [74, 83, 88]. In Step 4.4, a bipartition can be found in time $O(|E(B)|)$ by depth-first-search. The reduction of Definition 9 can be implemented in time $O(E)$ because the sets T^t, $X^\triangle$, and $Y^\triangle$ can easily be identified and removed. The most expensive task of relocating the edges of $E(N^\triangle, N_X^\triangle)$, $E(Y^\triangle, N_Y^\triangle)$, and $E(T^t, N^t)$ can be done by considering every edge of G and exchange one of its endpoints, if necessary. Hence, Step 4.5 can be done in time $O(|E(B)|)$.

Since G has at most $|V|$ blocks and, by Fact 1, the accumulated size of the blocks is at most two times the size of G, the accumulated runtime of Step 4 is in $O(|V| \cdot |E|)$. Furthermore, $|V|$ and $|V'|$ as well as $|E|$ and $|E'|$ differ in a constant factor only.

Lemma 53 guarantees that G' has an efficient edge dominating set of size Ω, if and only if G has an efficient edge dominating set of size Ω. Hence, Steps 4 and 5 are correct.

Step 5 can be done in time $O(|V'| + |E'|)$ by the algorithm for generalized series-parallel graphs given in [70], hence, in time $O(|V| + |E|)$.

The correctness of Step 6 follows from the proof of Lemma 53 and it can clearly be done in time $O(|V| \cdot |E|)$, as T^t, X, and Y are already calculated for every block in Steps 4.2 and 4.4. □

This completes the proof of Theorem 12.

As mentioned in the beginning of this section, the presented algorithm is robust, that is, if the input graph is not hole-free, the algorithm gives either a correct output or correctly states that the input graph is not hole-free. Recognizing hole-free graphs is an important problem. Besides a straightforward $O(|V|^5)$ algorithm proposed in [53], for a long time the best known algorithm was given in [89] with a runtime of $O(|V|^2 \cdot |V|^\alpha)$, where α is the exponent of multiplication of two $|V| \times |V|$ matrices. Today, the fastest known algorithm is given in [81] and has a runtime of $O(|V| + |E|^2)$.

One may ask whether hole-free graphs can be recognized with a modification of the presented algorithm in time $O(|V| \cdot |E|)$. Actually, this is not straightforwardly possible because the parts in which holes are found require the absence of induced K_4 and gems. In general hole-free graphs, there is no obvious way to reduce the K_4 or the gem without accidentally removing existing holes. Furthermore, our algorithm only finds the holes that are involved in the reduction of mandatory triangle edges.

4 Polarity and Monopolarity

The first part of this chapter gives three tractability results for MONOPOLARITY. These results build on one another. The first one, given in Section 4.1, describes a graph class that allows reducing MONOPOLAR EXTENSION to 2-SATISFIABILITY such that every monopolar extension of the input graph coincides with a satisfying truth assignment of the corresponding formula. The second one, given in Section 4.2, uses preprocessing to extend the first results to a larger graph class. This class contains all P_5-free graphs and all chair-free graphs and, hence, all claw-free graphs. The third result, given in Section 4.3, generalizes an approach for solving MONOPOLARITY on chordal graphs [36]. It shows that MONOPOLAR EXTENSION is efficiently solvable on graphs whose blocks are in the graph class that was given in Section 4.2. This yields a graph class that contains all hole-free graphs and, in particular, most graph classes for which efficient MONOPOLARITY algorithms are known.

Section 4.4 deals with polar graphs. Since POLARITY as well as MONOPOLARITY is NP-complete on planar graphs, we inspect subclasses of planar graphs. We show that POLARITY is efficiently solvable on maximal planar graphs, hole-free planar graphs, and chair-free planar graphs. These results build on the MONOPOLARITY-results of Sections 4.1 to 4.3.

This work is published in the extended abstract [64] and the article [63]. However, the results presented here are improved versions. In particular, the best tractability result for MONOPOLARITY shown here is slightly better than the MONOPOLARITY result recently given in [27].

Notions and Observations

The *paw*(v, w, x, y) consists of the triangle v–w–x–v and the vertex y which is adjacent only to x. The *diamond*(v, w, x, y) consists of the two triangles x–v–w–x and y–v–w–y. Figure 4.1 shows these two graphs.

A graphs is *maximal planar*, if it is planar and no edge can be added without loosing planarity. Those graphs are also called *triangulated planar*.

Figure 4.1: The graphs $paw(v, w, x, y)$ and $diamond(v, w, x, y)$.

Let $G = (V, E)$ be a graph, let $\mathcal{Q}$ be the set containing all blocks of G, and let C be the set containing all cutvertices of G. The *block-cutvertex-tree* T is a tree with node set $\mathcal{Q} \cup C$ and a block node $Q \in \mathcal{Q}$ is adjacent to a cutvertex node $c \in C$, if and only if $c \in V(Q)$. Let T be rooted at some block node $R \in \mathcal{Q}$. A node $x \in \mathcal{Q} \cup C$ is called *parent* of a node $y \in \mathcal{Q} \cup C$, if x and y are adjacent in T and $\mathrm{dist}_T(R, x) = \mathrm{dist}_T(R, y) - 1$. Consequently, all nodes with the same parent x are called *children* of x and the children of the children of x are called *grandchildren* of x. By T_x we denote the subtree of T rooted at the node x and by $G[T_x]$ we denote the subgraph of G that contains all blocks of T_x.

A path x–y–z in a graph G is called 3-path, if $\deg_G(x) = \deg_G(y) = \deg_G(z) = 2$.

Observation 55. *Let G be a graph with precoloring (A', B') and let x be a vertex of degree at most 1 that is not precolored. The graph G is (A', B')-monopolar extendable, if and only if $G - x$ is (A', B')-monopolar extendable.*

Proof. For the non-trivial direction, let (A, B) be a monopolar extension of (A', B') and $G - x$. If $\deg(x) = 0$, both, $(A, B \cup \{x\})$ and $(A \cup \{x\}, B)$ clearly are (A', B')-monopolar extensions of G. If $\deg(x) = 1$, let y be the neighbor of x in G. If $y \in A$, then $(A, B \cup \{x\})$ clearly is a monopolar extension of (A', B') and G and, if $y \in B$, then $(A \cup \{x\}, B)$ clearly is an (A', B')-monopolar extension of G. $\square$

Observation 56. *Let G be a graph with precoloring (A', B') and let x–y–z be a 3-path of G with no precolored vertex. The graph G is (A', B')-monopolar extendable, if and only if $G - \{x, y, z\}$ is (A', B')-monopolar extendable.*

Proof. For the non-trivial direction, let x' and z' be the neighbors of x and z respectively, that are different from y. For the non-trivial direction, let (A, B) be an (A', B')-monopolar extension of $G - \{x, y, z\}$. If $\{x', z'\} \subseteq A$, then $(A \cup \{y\}, B \cup \{x, z\})$ clearly is an (A', B')-monopolar extension of G. If $\{x', z'\} \subseteq B$, then $(A \cup \{x, z\}, B \cup \{y\})$ clearly is an (A', B')-monopolar extension of G. Hence, assume without loss of generality that $x' \in A$ and $z' \in B$. Clearly, $(A \cup \{z\}, B \cup \{x, y\})$ is an (A', B')-monopolar extension of G. □

Observation 57. *A graph G is (A', B')-monopolar extendable, if and only if A' is an independent set and $G - A'$ is $(\emptyset, B' \cup N(A'))$-monopolar extendable.*

Proof. Let (A, B) be a monopolar extension of (A', B') and G. Clearly, A is an independent set. Since $A' \subseteq A$ and A is independent, all vertices of $N(A')$ are in B. Since $G - A'$ is an induced subgraph of G that contains no vertex of A', $(A \setminus A', B)$ is a monopolar extension of $(\emptyset, B' \cup N(A'))$ and $G - A'$.

Conversely, let A' be an independent set and let (A, B) be a monopolar extension of $(\emptyset, B' \cup N(A'))$ and $G - A'$. Since A' is independent and $N(A') \subseteq B$, the set $A \cup A'$ is independent in G. Every induced P_3 of G that is not in $G - A'$ contains at least one vertex of A', hence, $(A \cup A', B)$ is a monopolar extension of (A', B') and G. □

A graph $G = (V, E)$ with precoloring (A', B') is called *simplified*, if $A' = \emptyset$, there is no vertex $y \in V \setminus B'$ with $\deg_G(y) \leq 1$, and there is no 3-path $x{-}y{-}z$ in G with $\{x, y, z\} \cap B' = \emptyset$.

4.1 A 2-satisfiability Approach for Monopolar Extension

This section introduces a graph class that admits a solution of MONOPOLAR EXTENSION by a reduction to a 2-SATISFIABILITY instance. 2-SATISFIABILITY is the restriction of SATISFIABILITY to 2-cnf input formulas, that is, formulas in conjunctive normal form with at most 2 literals per clause. While SATISFIABILITY and many of its restrictions are NP-complete, for example the problem mentioned in Section 5.1.1, 2-SATISFIABILITY is solvable in linear time [2, 34, 40].

The idea behind our reduction is as follows: A partition $V = A \cup B$ is a monopolar partition of a graph $G = (V, E)$, if and only if A is an independent set of G and contains at least one vertex of every induced P_3 of G. Hence, MONOPOLARITY can easily be reduced to SATISFIABILITY, in particular to 3-SATISFIABILITY, by constructing a boolean formula that has the vertices of the input graph as variables, the clause $(\neg x \vee \neg y)$ for every edge xy, and the clause $(x \vee y \vee z)$ for every induced P_3 $x{-}y{-}z$. It is easy to check that every satisfying truth assignment of the accordingly defined formula coincides with a monopolar partition of G. If we had the promise that a vertex x of an induced P_3 P was colored `blue` in every monopolar partition, that is, at least one of the other two vertices of P is colored `amber`, then the literal x of the clause that corresponds to P can be omitted. Hence, if we were given such a promise for at least one vertex of every induced P_3 of G, then MONOPOLARITY would be reducible to 2-SATISFIABILITY for this graph. Moreover, it is easy to implement a precoloring (A', B') into this formula by adding the clause (x) for every `amber` precolored vertex $x \in A'$ and the clause $(\neg y)$ for every `blue` precolored vertex $y \in B'$. This motivates the following definitions:

Definition 10. *A pair x, y of vertices of a graph G is called* required*, if for every monopolar partition (A, B) of G we have $\{x, y\} \cap A \neq \emptyset$.*

Definition 11. *An induced P_3 $x{-}y{-}z$ of a graph G is called* 2SAT-capable*, if at least one pair of its vertices is required. A graph G is called* 2SAT-capable*, if every induced P_3 of G is* 2SAT-capable*.*

Definition 12. *Let $G = (V, E)$ be graph with precoloring (A', B') and let $\mathcal{P}$ be the set that contains all required pairs of G. We define the boolean formula $F(G, A', B')$ in conjunctive normal form as follows:*

- *For every edge $xy \in E$, $F(G, A', B')$ contains the clause $(\neg x \vee \neg y)$.*
- *For every required pair $\{x, y\} \in \mathcal{P}$, $F(G, A', B')$ contains the clause $(x \vee y)$.*
- *For every $x \in A'$, $F(G, A', B')$ contains the clause (x).*
- *For every $x \in B'$, $F(G, A', B')$ contains the clause $(\neg x)$.*

Lemma 58. *A 2SAT-capable graph G with precoloring (A', B') admits a monopolar partition that extends (A', B'), if and only if $F(G, A', B')$ is satisfiable.*

Proof. For the first direction, let (A, B) be an (A', B')-monopolar extension of $G = (V, E)$. We define the truth assignment $b : V \to \{0, 1\}$ of $F(G, B')$ as follows: For every $v \in A$, let $b(v) = 1$ and, for every $v \in B$, let $b(v) = 0$. It is well defined because $A \cup B$ is a partition of V. Since A is an independent set in G, for every edge $xy \in E$, the clause $(\neg x \vee \neg y)$ is satisfied by b. By definition, for every required pair x, y, we have $\{x, y\} \cap A \neq \emptyset$, hence, the clause $(x \vee y)$ is satisfied by b. Since $A' \subseteq A$ and $B' \subseteq B$, the clause (x) is satisfied for every $x \in A'$ and the clause $(\neg y)$ is satisfied for every $y \in B'$. Hence, b is a satisfying truth assignment of $F(G, A', B')$.

For the other direction, let $b : V \to \{0, 1\}$ be a satisfying truth assignment of $F(G, A', B')$. We define the partition $V = A \cup B$ as follows: A contains every vertex $v \in V$ with $b(v) = 1$ and B contains every vertex $v \in V$ with $b(v) = 0$. For every edge $xy \in E$, the clause $(\neg x \vee \neg y)$ guarantees that at most one endpoint of xy is in A, hence, A is an independent set. Since G is 2SAT-capable, every induced P_3 of G is 2SAT-capable, that is, every induced P_3 contains a required pair. For every required pair x, y, the clause $(x \vee y)$ guarantees that x, y, or both are in A. Hence, $G[B]$ is P_3-free. For every $x \in A'$ and every $y \in B'$, the clauses (x) and $(\neg y)$ guarantee that $x \in A$ and $y \in B$, hence, (A, B) is an (A', B')-monopolar extension of G. □

This implies:

Corollary 59. *Let $\mathcal{C}$ be a subclass of the 2SAT-capable graphs. If the required pairs of a graph $G = (V, E) \in \mathcal{C}$ with precoloring (A', B') can be computed in time $T(|G|, |A'|, |B'|)$, then* MONOPOLAR EXTENSION *can be solved in time $O(|V| + |E| + T(|G|, |A'|, |B'|))$ on G.*

Proof. First, we compute the set $\mathcal{P}$ of required pairs of G and B'. This takes time $T(|G|, |A'|, |B'|)$, hence, we have $|\mathcal{P}| \leq T(|G|, |A'|, |B'|)$. Clearly, constructing $F(G, A', B')$ takes at most time $O(|E|+|\mathcal{P}|+|A'|+|B'|)$ because we have a clause for every edge, every required pair, and every precolored vertex. Since G is 2SAT-capable, we can solve MONOPOLAR EXTENSION on G and (A', B') by solving 2-SATISFIABILITY on $F(G, A', B')$ by Lemma 58. Since $F(G, A', B')$ has at most $|E| + |\mathcal{P}| + |A'| + |B'|$ clauses and 2-SATISFIABILITY is linear time solvable [2, 34, 40], this can be done in time $O(|E| + |\mathcal{P}| + |A'| + |B'|) \subseteq O(|V| + |E| + T(G, A', B'))$. □

We establish a graph class in this section that is a subclass of the 2SAT-capable graphs. We use the following observations to describe configurations that imply required pairs:

Observation 60. *If a graph G contains an induced paw(v, w, x, y) or an induced diamond(v, w, x, y), then the pair x, y is required.*

Proof. Assume that the pair x, y is not required, that is, there is a monopolar partition (A, B) of G with $\{x, y\} \cap A = \emptyset$. If G contains an induced *paw*(v, w, x, y), then $v \in A$ because otherwise $G[B]$ contains the induced P_3 v–x–y. Since A is independent, we have $w \notin A$ and w–x–y is induced in

$G[B]$—this is a contradiction. If G contains an induced $diamond(v, w, x, y)$, then $v \in A$ because otherwise $G[B]$ contains the induced P_3 $x{-}v{-}y$. Since A is independent, we have $w \notin A$ and $x{-}w{-}y$ is induced in $G[B]$—this is a contradiction. □

Observation 61. *If a graph G contains an induced cycle $v{-}w{-}x{-}y{-}v$, then the pair v, w, the pair w, x, the pair x, y, and the pair v, y is required.*

Proof. Assume that the pair x, y is not required, that is, there is a monopolar partition (A, B) of G with $\{x, y\} \cap A = \emptyset$. If there is an induced C_4 $v{-}w{-}x{-}y{-}v$ in G, then $v \in A$ because otherwise $x{-}y{-}v$ is an induced P_3 in $G[B]$. Since A is independent, we have $w \notin A$ and $w{-}x{-}y$ is induced in $G[B]$—this is a contradiction. The observation follows by symmetry. □

Observation 62. *If a vertex x of a graph $G = (V, E)$ is* `blue` *precolored, then every pair $\{y, z\} \subseteq N(x)$ with $y \cdots z$ and every pair y, z with $y \in N(x)$, $z \in N^2(x)$, and $y{-}z$ is required.*

Proof. Let $G = (V, E)$ be a graph with precoloring (A', B') and let $x \in B'$. Assume that the pair y, z is not required, that is, there is an $(A'B')$-monopolar extension (A, B) of G with $\{y, z\} \cap A = \emptyset$. If $\{y, z\} \subseteq N^1(x)$ and $y \cdots z$, then $y{-}x{-}z$ is an induced P_3 in $G[B]$—this is a contradiction. If $y \in N^1(x)$, $z \in N^2(x)$, and $y{-}z$, then $x{-}y{-}z$ is an induced P_3 in $G[B]$—this is a contradiction. □

Using these observations, we can define a class of graphs that are 2SAT-capable:

Definition 13. *For a graph $G = (V, E)$ with precoloring (A', B'), an induced P_3 P is called* defused*, if one of the following holds:*

- *A vertex of P is part of a triangle of G.*
- *An edge of P is part of an induced C_4 of G.*
- *A vertex of P is* `blue` *precolored of G.*
- *A vertex of P is adjacent to a* `blue` *precolored vertex of G.*

*We say that (G, A', B') is P_3-*defused*, if every induced P_3 of G is defused. A graph G without precoloring is called P_3-*defused*, if $(G, \emptyset, \emptyset)$ is P_3-defused.*

Lemma 63. *Every defused P_3 contains at least one required pair, hence, every P_3-defused (precolored) graph is 2SAT-capable.*

Proof. Let $G = (V, E)$ be a graph and let $P = x{-}y{-}z$ be an induced defused P_3 of G, that is, P contains a vertex of a triangle or an edge of an induced C_4 or a vertex of $N[V(P)]$ is `blue` precolored.

Assume that a vertex of P is part of a triangle $T = u{-}v{-}w{-}u$. Clearly, P and K cannot share more than two vertices. First, assume that P shares two vertices, that is, an edge with T, say $u = x$ and $v = y$. If $w \cdots z$, then $paw(x, w, y, z)$ is induced in G and, by Observation 60, y, z is a required pair. Otherwise, if $w{-}z$, then $diamond(w, y, x, z)$ is induced in G and, by Observation 60, x, z is a required pair. Now assume that P shares a single vertex with T, say $x = u$. If $v{-}y$ or $w{-}y$, then $T' = u{-}v{-}y{-}u$ or $T'' = u{-}w{-}y{-}u$ is a triangle that shares an edge with P, and, as already shown, P contains a required pair. If $v \cdots y$ and $w \cdots y$, then $paw(v, w, x, y)$ is induced in G and, by Observation 60, u, y is a required pair.

Assume that an edge of P, say xy, is part of an induced C_4. By Observation 61, x, y is a required pair.

Assume that a vertex of P is `blue` precolored. If x is `blue` precolored, then $y \in N^1(x)$ and $z \in N^2(x)$ with $y{-}z$ and, by Observation 62, y, z is a required pair. Analogously, if z is `blue` precolored, then x, y is a required pair. If y is `blue` precolored, then $\{x, z\} \subseteq N^1(x)$ with $x \cdots z$ and, by Observation 62, x, z is a required pair.

Assume that a vertex of P is adjacent to a `blue` precolored vertex v. If $v \in V(P)$, we are in the previous case, hence, let $v \notin V(P)$. First, assume that v is adjacent to exactly one vertex of P. If $v{-}x$, then $x \in N^1(v)$ and $y \in N^2(v)$ with $x{-}y$ and, by Observation 62, x, y is a required pair. Analogously, if $v{-}z$, then y, z is a required pair and if $v{-}y$, then x, y and y, z are required pairs. Now, assume that v is adjacent to exactly two vertices of P. If $v{-}x$ and $v{-}z$, then $v{-}x{-}y{-}z{-}v$ is induced, hence, the edges of P are part of an induced C_4 and, as already shown, P contains a required pair. If $v{-}x$ and $v{-}y$ or $v{-}y$ and $v{-}z$, then P shares an edge with a triangle and, as already shown, P contains a required pair. Finally, assume that v is adjacent to three vertices of P. Again, P shares an edge with a triangle and, hence, contains a required pair.

This shows that P contains at least one required pair, that is, P is 2SAT-capable. Clearly, if every induced P_3 is defused, then G is 2SAT-capable. $\square$

Finally, we show that the required pairs of P_3-defused graphs can be found efficiently.

Observation 64. *The required pairs described by Observations 60 to 62 of a P_3-defused graph $G = (V, E)$ with precoloring (A', B') can be found in time*

$O(|V|^4)$.

Proof. Clearly, all induced diamonds, all induced paws, and all induced C_4 of G can be found by considering every subset of V with four elements, hence, the required pairs of Observations 60 and 61 can be found in time $O(|V|^4)$. The required pairs of Observation 62 can be found in time $O(|B'| \cdot |V|^2) \subseteq O(|V|^3)$ by starting a breadth-first-search on every vertex x of B' to determine $N^1(x)$ and $N^2(x)$ and checking for every pair of vertices if it fulfills the necessary conditions. □

The class of P_3-defused graphs might look quite unnatural, but it includes some well-known classes, for example all biconnected hole-free graphs and all maximal planar graphs:

Lemma 65. *Biconnected hole-free graphs and maximal planar graphs are P_3-defused.*

Proof. Let G be a biconnected hole-free graph and let $P = x{-}y{-}z$ be an induced P_3 in G. Let $x{-}v_1{-}v_2{-}\ldots{-}v_p{-}z$ be a shortest path between x and z in $G - y$. Since G is biconnected, such a path exists and, since $x \cdots z$, we have $p \geq 1$. If $p = 1$, then $\{v_1, x, y, z\}$ induces either a diamond or a C_4 in G, hence, P is defused. Thus, assume that $p \geq 2$. If $y{-}v_1$, then $v_1{-}x{-}y{-}v_1$ is a triangle and, hence, P is defused. If $y \cdots v_1$ but $y{-}v_2$, then $v_1{-}v_2{-}y{-}x{-}v_1$ is an induced C_4 in G and, hence, P is defused. If $y \cdots y_1$ and $y \cdots y_2$, then G clearly contains an induced hole—this is a contradiction. This shows that G is P_3-defused.

It is well known that in a maximal planar graph with at least 4 vertices every edge belongs to at least two triangles. Hence, every induced P_3 of a maximal planar graph is defused. □

Corollary 59, Lemma 65, and Observation 64 together imply:

Theorem 13. MONOPOLAR EXTENSION *and, hence,* MONOPOLARITY *can be solved on P_3-defused graphs and, hence, on biconnected hole-free graphs and maximal planar graphs in time $O(n^4)$, where n is the number of vertices of the input graph.*

Notice that every satisfying truth assignment of $F(G, A', B')$ coincides with an (A', B')-monopolar extension of G. Hence, this reduction not only solves the decision version of the problem but also the search version.

Section 4.2 uses the technique presented here to show tractability of MONOPOLAR EXTENSION on a superclass of chair-free graphs and Section 4.3

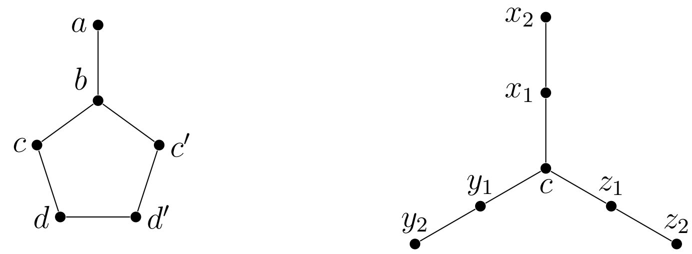

Figure 4.2: The graphs $A_5(a, b, c, c', d, d')$ and $S_{2,2,2}(c, x_1, x_2, y_1, y_2, z_1, z_2)$.

extends this result even further to a graph class that contains all hole-free graphs. Although the algorithms in the following two sections are potentially able to solve the search problem, we have to consider simplified graphs only and, hence, the one-to-one relationship between the desired monopolar extensions and the satisfying truth assignments of a 2-cnf is lost.

4.2 Monopolar Extension on a Superclass of Chair-free Graphs

The 5-*apple*, denoted by $A_5(a, b, c, c', d, d')$, is the graph consisting of an induced cycle $b{-}c{-}d{-}d'{-}c'{-}b$ and an additional vertex a that is adjacent only to b. The (i, j, k)-*star*, denoted by $S_{i,j,k}(c, x_1, \ldots, x_i, y_1, \ldots, y_j, z_1, \ldots, z_k)$, is the union of three induced paths, $c{-}x_1{-}\ldots{-}x_i$, $c'{-}y_1{-}\ldots{-}y_j$, and $c''{-}z_1{-}\ldots{-}z_k$, on $c = c' = c''$. The graph $(1, 1, 1)$-star is also called *claw* and the graph $(2, 1, 1)$-star is also called *chair*. Figure 4.2 shows the 5-apple and the $(2, 2, 2)$-star.

Definition 14. *An induced* 5*-apple* $A_5(a, b, c, c', d, d')$ *is called* defused*, if*

$$a{-}b{-}c, \quad a{-}b{-}c', \quad \textit{and} \quad c{-}b{-}c'$$

are defused. An induced $(2, 2, 2)$*-star* $S_{2,2,2}(c, x_1, x_2, y_1, y_2, z_1, z_2)$ *is called* defused*, if*

$$x_1{-}c{-}y_1, \quad x_1{-}c{-}z_1, \quad \textit{and} \quad y_1{-}c{-}z_1$$

are defused. A graph is called A_5-$S_{2,2,2}$-defused*, if every induced* 5*-apple and every induced* $(2, 2, 2)$*-star is defused.*

In this section, we show that MONOPOLAR EXTENSION is efficiently solvable on the class of A_5-$S_{2,2,2}$-defused graphs. Notice that every $\{A_5, S_{2,2,2}\}$-free

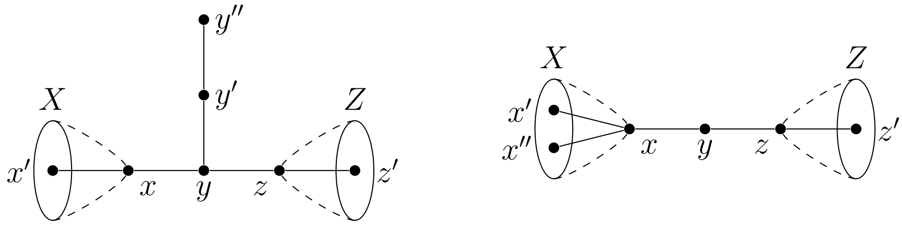

Figure 4.3: A proof sketch for Observation 67 and Lemma 68.

graph is A_5-$S_{2,2,2}$-defused. Furthermore, since the 5-apple and the $(2, 2, 2)$-star contain the P_5, the chair and the claw as induced subgraphs, this is a superclass of P_5-free graphs, chair-free graphs, and claw-free graphs.

The algorithm in this section stepwise reduces MONOPOLAR EXTENSION on A_5-$S_{2,2,2}$-defused graphs to MONOPOLAR EXTENSION on P_3-defused graphs. We restrict the input to simplified graphs, so we apply Observations 55 to 57 when necessary.

For an induced P_3 of the form $P = x–y–z$, we define

$$X := N(x) \setminus \{y\} \text{ and } Z := N(z) \setminus \{y\} .$$

As a prerequisite, we need the following two observations:

Observation 66. *Let G be a simplified A_5-$S_{2,2,2}$-defused graph with* `blue` *precolored vertices B' and let $P = x–y–z$ be an induced P_3 of G. If P is not defused, then X and Z are non-empty disjoint independent sets.*

Proof. As G is simplified, every vertex of $V \setminus B'$ has degree at least 2. Since P is not defused, no vertex of $N[V(P)]$ is precolored. Hence, X and Z are non-empty. If X or Z is not independent, then x or z respectively would be part of a triangle, that is, P would be defused. If X and Z share a vertex, say w, then $w–x–y–z–w$ would be induced, that is, P would be defused. Hence, X and Z are disjoint and $G[X]$ and $G[Z]$ are edgeless. □

Observation 67. *Let G be a simplified A_5-$S_{2,2,2}$-defused graph with* `blue` *precolored vertices B' and let $P = x–y–z$ be an induced P_3 of G. If P is not defused, then* $\deg(y) = 2$.

Proof. Figure 4.3 shows a sketch for this proof. Conversely, assume that P is not defused and $\deg(y) \geq 3$. Let $y' \notin \{x, z\}$ be a neighbor of y. Since $y' \in N[V(P)]$ and P is not defused, y' is not precolored and, since G is simplified, y' has degree at least 2. Consequently, let y'' be a neighbor of y'

that is different from y. Since P is not defused, it shares no vertex with a triangle and no edge with an induced C_4, so y' is not adjacent to any vertex in $\{x,z\} \cup X \cup Z$ and we have $x \cdots y''$, $y \cdots y''$, and $z \cdots y''$. In particular, this means $y' \notin X \cup Z$ and $y'' \notin X \cup Z$. If there are vertices $x' \in X$ and $z' \in Z$ with $x' - z'$, then $A_5(y', y, x, z, x', z')$ is induced in G. Since G is A_5-defused, this implies that $x - y - z$ is defused—this is a contradiction. Hence, we have $E(X, Z) = \emptyset$. If y'' has a neighbor $w \in X \cup Z$, then $A_5(x, y, y', z, y'', w)$ or $A_5(z, y, y', x, y'', w)$ is induced in G. Again, since G is A_5-defused, this implies that $x - y - z$ is defused—this is a contradiction. Hence, we have $E(y'', X \cup Z) = \emptyset$. Let $x' \in X$ and $z' \in Z$. By Observation 66 and $E(X, Z) = E(y'', X \cup Z) = \emptyset$, x' and z' exist and $S_{2,2,2}(y, x, x', y', y'', z, z')$ is induced in G. Finally, since G is $S_{2,2,2}$-defused, this implies that $x - y - z$ is defused—this is a contradiction. $\square$

One step of the reduction is given by the following lemma:

Lemma 68. *Let G be a simplified A_5-$S_{2,2,2}$-defused graph with* `blue` *precolored vertices B' and let $P = x - y - z$ be an induced P_3 of G. If P is not defused, then G is $(\emptyset, B')$-monopolar extendable, if and only if $G - y$ is $(\emptyset, B')$-monopolar extendable.*

Proof. Figure 4.3 shows a sketch for this proof. For the non-trivial direction, let (A, B) be an $(\emptyset, B')$-monopolar extension of $G - y$. Assume that P is not defused. By Observation 67, we have $\deg(y) = 2$ and, since G contains no 3-path of not precolored vertices, we have $|X| > 1$ or $|Z| > 1$. Without loss of generality, assume that $|X| > 1$ and let $x' \in X$ be chosen arbitrarily. By definition of X, we have $x' \cdots y$. This means that $x' - x - y$ is an induced P_3 in G. Since $|X| > 1$, there is at least one vertex $x'' \in X$ that differs from x', that is, we have $\deg(x) \geq 3$. Thus, Observation 67 implies that $x' - x - y$ is defused, that is, the pair x', x or the pair x', y or the pair x, y is required. If x, y is a required pair, then P is defused—this is a contradiction. If x', y is a required pair, then, by Observations 60 and 62, we have $\{x', y\} \subseteq N(w)$ for some $w \in B'$ or G contains an induced $diamond(x, v, x', y)$ for some $v \in V$. If $\{x', y\} \subseteq N(w)$ for some $w \in B'$, then P has the neighbor w in B' and, hence, P is defused—this is a contradiction. If G contains the induced $diamond(x, v, x', y)$ for some $v \in V$, then at least two vertices of P are part of a triangle and, hence P is defused—this is a contradiction. Thus, x', x is a required pair. Since x' was chosen arbitrarily from X, this implies for every $x' \in X$ that x, x' is a required pair, that is, $x' \in A \Leftrightarrow x \in B$. Analogously, if additionally $|Z| > 1$, then for every $z' \in Z$ we have $z' \in A \Leftrightarrow z \in B$.

If $\{x, z\} \subseteq A$, then $(A, B \cup \{y\})$ clearly is an $(\emptyset, B')$-monopolar extension of G. If $\{x, z\} \subseteq B$, then $(A \cup \{y\}, B)$ clearly is an $(\emptyset, B')$-monopolar extension

Algorithm: A_5-$S_{2,2,2}$`-defused-Reduction`
Input: An A_5-$S_{2,2,2}$-defused graph $G = (V, E)$ with precoloring (A', B').
Output: A P_3-defused graph G' such that G is (A', B')-monopolar extendable, if and only if G' is $(\emptyset, B' \cup N(A'))$-monopolar extendable.

1. Check if A' is independent in G. If not, then `Return` some trivial no-instance.
2. Determine $N(A')$ and set $G := G - A'$ and $B' := B' \cup N(A')$.
3. Determine the set $\mathcal{P}$ of required pairs defined by Observations 60 to 62.
4. Do
   1. If G contains a vertex $y \notin B'$ with $\deg(v) \leq 1$, then set $G := G - y$,
   2. else, if G contains a 3-path $x - y - z$ with $\{x, y, z\} \cap B' = \emptyset$, then set $G := G - \{x, y, z\}$,
   3. else, if G contains an induced P_3 of G, say $x - y - z$, with $\deg(y) = 2$ that contains no pair of $\mathcal{P}$, then set $G := G - y$.
   4. Remove every pair from $\mathcal{P}$ that contains a vertex that is not in G.

   until G was not changed in the last iteration.
5. `Return` G.

Table 4.1: Reduction from MONOPOLAR EXTENSION on A_5-$S_{2,2,2}$-defused graphs to MONOPOLAR EXTENSION on P_3-defused graphs.

of G. If $x \in B$ and $z \in A$, then $(A, B \cup \{y\})$ is an $(\emptyset, B')$-monopolar extension of G because, as shown above, $x \in B$ implies $X \subseteq A$. Analogously, if $x \in A$ and $z \in B$ and $|Z| > 1$, then $(A, B \cup \{y\})$ is an $(\emptyset, B')$-monopolar extension of G.

Hence, assume that $x \in A$, $z \in B$, and $|Z| = 1$ with $Z = \{z'\}$. If $z' \in A$, then $(A, B \cup \{y\})$ clearly is an $(\emptyset, B')$-monopolar extension of G. If $z' \in B$, then $(A \cup \{z\}, B \setminus \{z\} \cup \{y\})$ is an $(\emptyset, B')$-monopolar extension of G because $z' \in B$ is the only neighbor of z in $G - y$ and, therefore, recoloring z from `blue` to `amber` keeps A independent. □

With the foregoing lemma, we can formulate the recursive reduction algorithm given in Table 4.1.

Lemma 69. *The algorithm* A_5-$S_{2,2,2}$`-defused-Reduction` *is correct and runs in time* $O(n^4)$ *on input graphs with* n *vertices.*

Proof. Steps 1 and 2 are justified by Observation 57. Since testing A' for being independent in G takes at most time $O(|E|) \subseteq O(|V|^4)$, $N(A')$ can be determined in linear time by breadth-first-search, and constructing $G - A'$ takes at most time $O(|G|) \leq O(|V|^4)$, both steps can be done in time $O(|V|^4)$. In Step 3, the set of required pairs can be found in time $O(|V|^4)$ by Observation 64.

The correctness of the Steps 4.1 and 4.2 follows from Observations 55 and 56. Step 4.3 is executed only if G contains no not precolored vertex of degree 1 and no not precolored 3-path, that is, if G is simplified. Hence, its correctness is given by Lemma 68.

After the execution of one of the Steps 4.1 to 4.3, the set $\mathcal{P}$ shall contain all required pairs defined by Observations 60 to 62 of the reduced graph. The deletion of one or more vertices of a graph cannot introduce new required pairs because no new induced paws, diamonds, or C_4 can arise and, for every `blue` precolored vertex x, the sets $N^1(x)$ and $N^2(x)$ cannot increase. Hence, we do not have to add new elements to $\mathcal{P}$. We show that every required pair of G that does not contain one of the removed vertices is still a required pair in the reduced graph:

Assume that a required pair x, y results from an induced $paw(v, w, x, y)$ or an induced $diamond(v, w, x, y)$. The vertices v and w have degree at least 2. But we remove a vertex of degree 2 only if it is the midpoint of a P_3. This is not the case for v and w. Hence, if the reduced graph contains x and y, then it also contains $paw(v, w, x, y)$ or $diamond(v, w, x, y)$ as induced subgraph.

Now assume that a required pair x, y results from an induced cycle $v–w–x–y–v$. Again, the vertices v and w have degree at least 2. If $v–w–x$ or $w–x–y$ or $x–y–v$ or $y–v–w$ is a 3-path in G without precolored vertices, then we remove at least one of x and y. Since vw is an edge of an induced C_4, neither v nor w can be of degree 2 and be the midpoint of a P_3 that contains no pair of $\mathcal{P}$, that is, neither v nor w are removed in that case.

Finally, assume that a required pair x, y results from a `blue` precolored vertex that is adjacent to one or two vertices of the pair. Since we do not remove precolored vertices, the required pair also exists in the reduced graph, unless a vertex of the required pair is removed. This shows that the reduced graph contains exactly the required pairs of G except pairs that contain one of the removed vertices. Hence, Step 4.4 is correct.

The loop in Step 4 is executed at most $|V|$ times because at least one vertex of G is deleted in every iteration. We show that the execution of the body of the loop can be done in time $O(|V|^3)$: Whenever the body is executed, we first count the degree of every vertex of G. This clearly can be done in time $O(|V| + |E|) \subseteq O(|V|^2)$. To check the condition of Step 4.1, we simply consider every vertex of degree 1 and check if it is not in B', which can be done in time $O(|V|)$. To check the condition of Step 4.2, we consider every vertex of degree 2 that is not in B' and check if its two neighbors are of degree 2 and not in B'. Clearly, this can also be done in time $O(|V|)$. To check the condition of Step 4.3, we consider every vertex y of degree 2, check if its two neighbors x and z are not adjacent and if $\mathcal{P}$ contains no pair of $\{x, y, z\}$. This can be done in time $O(|V|+|\mathcal{P}|) \subseteq O(|V|^2)$. Constructing $G-y$, respectively $G-\{x, y, z\}$, can obviously be done in time $O(|G|) \subseteq O(|V|^2 \log |V|) \subseteq O(|V|^3)$. Step 4.4 clearly can be done in time $O(|\mathcal{P}|) \subseteq O(|V|^2)$.

When the loop finishes, G is simplified and every P_3 of G, whose midpoint is of degree 2, is defused. Hence, Observation 67 implies that every induced P_3 of G is defused, that is, G is P_3-defused and Step 5 is correct. □

Theorem 13 and Lemma 69 together imply:

Theorem 14. Monopolar extension *and, hence,* monopolarity *is solvable in time $O(n^4)$ on A_5-$S_{2,2,2}$-defused graphs and, hence, on $\{A_5, S_{2,2,2}\}$-free graphs, P_5-free graphs, chair-free graphs, and claw-free graphs, where n is the number of vertices of the input graph.*

4.3 Monopolar Extension on a Superclass of Hole-free Graphs

In this section, we define *locally A_5-$S_{2,2,2}$-defused graphs*, that is, graphs whose blocks are A_5-$S_{2,2,2}$-defused. We show that monopolar extension is polynomial-time solvable on this graph class using a technique that can be seen as a generalization of the linear time algorithm for chordal graphs given in [36].

The largest graph class for which monopolarity is known to be tractable is the class $\mathcal{G}$ defined in [27]. A graph is in $\mathcal{G}$, if every induced P_3 $x-y-z$ with $\deg(y) \geq 3$ that is wholly contained in a cycle of G contains a vertex of a triangle or an edge of an induced C_4. Churchley and Huang show that the class $\mathcal{G}$ contains many well-studied graphs classes like hole-free graphs and claw-free graphs and, hence, cographs, chordal graphs, permutation graphs and co-comparability graphs. The following observation shows that $\mathcal{G}$ is a proper subclass of the locally A_5-$S_{2,2,2}$-defused graphs. Admittedly, the difference between the two classes is not substantial.

Observation 70. *All graphs in $\mathcal{G}$ are locally A_5-$S_{2,2,2}$-defused and there are infinitely many A_5-$S_{2,2,2}$-defused graphs that are not in $\mathcal{G}$.*

Proof. Let $G = (V, E) \in \mathcal{G}$ and assume that G is not locally A_5-$S_{2,2,2}$-defused. That is, there is a block B of G that contains an induced 5-apple or $(2, 2, 2)$-star that is not defused. Hence, there is an induced P_3 in G, say $x{-}y{-}z$, that is not defused such that G contains an induced $A(x, y, z, u, v, w)$ or $A(u, y, x, z, v, w)$ or $S_{2,2,2}(y, x, u, z, v, w, t)$ for some other vertices $\{u, v, w, t\} \subseteq V$. Since B is a block, that is, B is biconnected, $x{-}y{-}z$ is contained in a cycle. Furthermore, in all configurations, $\deg(y) \geq 3$. Hence, by definition of $\mathcal{G}$, $x{-}y{-}z$ shares a vertex with a triangle or an edge with an induced C_4, that is, $x{-}y{-}z$ is defused—this is a contradiction.

Locally A_5-$S_{2,2,2}$-defused graphs can contain an induced P_3 that is not defused whose midpoint is of degree at least 3 as long as one vertex of the P_3 is a cutvertex. Hence, there are infinitely many graphs that are locally A_5-$S_{2,2,2}$-defused but not in $\mathcal{G}$. As an example, consider all apples starting from the A_5, that is, the graphs consisting of an induced cycle of length at least 5 and an additional vertex that is adjacent to exactly one vertex of the cycle. The induced P_3 of the cycle such whose midpoint is adjacent to the additional vertex shares no vertex with a triangle and no edge with an induced C_4, hence, the graph is not in $\mathcal{G}$. The blocks of every apple are an induced cycle and a single edge. Hence, the blocks are $\{A_5, S_{2,2,2}\}$-free and, therefore, every apple is locally A_5-$S_{2,2,2}$-defused. □

The idea of this section is to solve MONOPOLAR EXTENSION on a locally A_5-$S_{2,2,2}$-defused graph G recursively using the block-cutvertex tree T rooted at some block node R. More precisely, we decide MONOPOLAR EXTENSION on the subgraph $G[T_Q]$ for some block Q of G in the following way: First, we solve three MONOPOLAR EXTENSION instances on $G[T_{Q'}]$ for every grandchild Q' of Q to gather information about the behavior, in all (A', B')-monopolar extensions of $G[T_{Q'}]$, of the cutvertex that connects Q and Q'. The information we gather is formalized in three properties, namely "forced `amber`", "forced `blue`", and "critical", which are defined later. After computing these properties for every child of Q, we construct a graph $Q_{\texttt{abc}}$ from Q by adding `blue` precolored pending vertices to the cutvertices of Q that are children of Q in T depending on the just determined properties of these cutvertices. The graph $Q_{\texttt{abc}}$ is (A', B')-monopolar extendable, if and only if $G[T_Q]$ is

(A', B')-monopolar extendable. Hence, to decide if G is (A', B')-monopolar extendable, we decide if $G = G[T_R]$ is (A', B')-monopolar extendable using recursion.

Before defining forced `amber`, forced `blue`, and critical and showing how to compute them, we show that this reduction is applicable on locally A_5-$S_{2,2,2}$-defused graphs. Although MONOPOLAR EXTENSION is efficiently solvable on A_5-$S_{2,2,2}$-defused graphs and, hence, on the blocks of a locally A_5-$S_{2,2,2}$-defused graph, this is not obvious because we modify the blocks of G. The modification is limited to add `blue` precolored vertices. Hence, we show:

Lemma 71. *Let $G = (V, E)$ be a A_5-$S_{2,2,2}$-defused graph with precoloring (A', B') whose vertices of degree 1 are all in B'. The graph $G' = (V \cup \{x\}, E \cup \{xy\})$ with precoloring $(A', B' \cup \{x\})$ for a new vertex $x \notin V$ and every vertex $y \in V$ is A_5-$S_{2,2,2}$-defused and all its vertices of degree 1 are in $B' \cup \{x\}$.*

Proof. For some $y \in V$, assume that G' is not A_5-$S_{2,2,2}$-defused. Since we only add vertices to G, one can easily check that every induced 5-apple and every induced $(2,2,2)$-star of G that is defused, is also defused in G'. Hence, there is an induced A_5 or induced $S_{2,2,2}$ that contains x and is not defused.

Assume that G' contains an induced 5-apple that is not defused. Since $\deg(x) = 1$, the 5-apple is of the form $A(x, y, c, d, c', d')$. By x is `blue` precolored, Definition 13 implies that $x{-}y{-}c$, $x{-}y{-}c'$, and $c{-}y{-}c'$ are defused—this is a contradiction to the assumption that the induced 5-apple is not defused.

Assume that G' contains an induced $(2,2,2)$-star that is not defused. Since $\deg(x) = 1$, the $(2,2,2)$-star is of the form $S_{2,2,2}(c, y, x, y', x', y'', x'')$. If $\deg_G(y) = 1$, what implies $y \in B'$, then, by Definition 13, $y{-}c{-}y'$, $y{-}c{-}y''$, and $y'{-}c{-}y''$ are defused—this is a contradiction. Hence, assume that $\deg_G(y) \geq 2$ and let w be a neighbor of y that is different from c. Figure 4.4 shows a sketch for this case. If $S_{2,2,2}(c, y, w, y', x', y'', x'')$ is induced in G, then $y{-}c{-}y'$, $y{-}c{-}y''$, and $y'{-}c{-}y''$ are defused because G is A_5-$S_{2,2,2}$-defused—this is a contradiction. Hence, w is adjacent to c, y', y'', x', or x''. If $w{-}c$, then $y{-}c{-}y'$, $y{-}c{-}y''$, and $y'{-}c{-}y''$ share at least one vertex with the triangle $w{-}y{-}c{-}w$ and, hence, are defused—this is a contradiction. If $w{-}y'$, then $y{-}c{-}y'$, $y{-}c{-}y''$, and $y'{-}c{-}y''$ share at least one edge with the induced C_4 $w{-}y{-}c{-}y'{-}w$ and, hence, are defused—this is a contradiction. Analogously, $w{-}y''$ yields a contradiction. If $w{-}x'$, then $y{-}c{-}y'$, $y{-}c{-}y''$, and $y'{-}c{-}y''$ are defused because $A_5(y'', c, y, y', w, x')$ is induced in G and, hence, defused—this is a contradiction. Analogously, $w{-}x''$ results in a contradiction. Hence, G' is A_5-$S_{2,2,2}$-defused and, clearly, every vertex of degree 1 is `blue` precolored. □

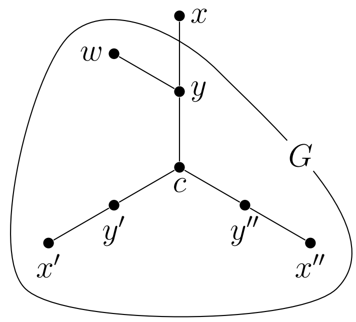

Figure 4.4: A sketch for the proof of Lemma 71 for the case that G' contains an induced $(2, 2, 2)$-star that is not defused and $\deg_G(y) \geq 2$.

Since the blocks of a locally A_5-$S_{2,2,2}$-defused graph G are A_5-$S_{2,2,2}$-defused and biconnected, that is, without vertices of degree 1, Lemma 71 applies on the blocks of G. Hence, adding `blue` precolored vertices to a block results in a graph that is A_5-$S_{2,2,2}$-defused and, hence, we can solve MONOPOLAR EXTENSION efficiently on it.

Forced `amber`, forced `blue` and critical are defined as follows:

Definition 15. *Let G be a graph with precoloring (A', B'). A vertex x of G is called* forced `amber`*, if it is colored* `amber` *in every (A', B')-monopolar extension of G and* forced `blue`*, if it is colored* `blue` *in every (A', B')-monopolar extension of G. A vertex is called* critical*, if it has at least one* `blue` *neighbor in every (A', B')-monopolar extension of G.*

Notice that G is (A', B')-monopolar extendable, if and only if there is no vertex in G simultaneously forced `amber` and forced `blue`. Actually, if G has no (A', B')-monopolar extension, then every vertex is forced `amber`, forced `blue`, and critical by vacuous truth.

As stated above, our algorithm is based on the algorithm for chordal graphs given in [36]. The key difference between the algorithm for chordal graphs and our algorithm is the need for the definition of critical vertices and interpreting this property during the recursion. In chordal graphs, every vertex of a non-trivial block is critical because it is part of a triangle and in triangles, at least two vertices are colored `blue` in every monopolar partition. This is not the case in locally A_5-$S_{2,2,2}$-defused graphs. For example, in the graph C_4, which clearly can be a non-trivial block of a locally A_5-$S_{2,2,2}$-defused graph, no vertex is critical. Furthermore, monopolar chordal blocks are split graphs, which allows an obvious MONOPOLAR EXTENSION algorithm.

Generally, we can decide for a vertex whether it is forced `amber`, forced `blue`, or critical by solving a MONOPOLAR EXTENSION instance:

Observation 72. *Let G be a graph with precoloring (A', B').*

- *A vertex x is forced* `amber`*, if and only if G is not $(A', B' \cup \{x\})$-monopolar extendable.*
- *A vertex x is forced* `blue`*, if and only if G is not $(A' \cup \{x\}, B')$-monopolar extendable.*
- *A vertex x is critical, if and only if G is not $(A' \cup N(x), B')$-monopolar extendable.*

Proof. If x is forced `amber`, that is, $x \in A$ and, hence, $x \notin B$, for every (A', B')-monopolar extension (A, B) of G, then G clearly is not $(A', B' \cup \{x\})$-monopolar extendable. If G is not $(A', B' \cup \{x\})$-monopolar extendable, that is, there is no (A', B')-monopolar extension (A, B) of G with $x \in B$, hence, in every (A', B')-monopolar partition (A, B) of G we have $x \in A$, then x is forced `amber`. An analogous argumentation works for x being forced `blue` or critical, respectively G being not $(A' \cup \{x\}, B')$-monopolar extendable or not being $(A' \cup N(x), B')$-monopolar extendable. □

Observation 72 enables us to check whether a cutvertex c of a block Q of G is forced `amber`, forced `blue`, or critical in $G[T_Q]$ by solving three MONOPOLAR EXTENSION instances on $G[T_Q]$. But we are interested if c is forced `amber`, forced `blue`, or critical in $G[T_c]$. This can be decided very easily with the following lemma:

Lemma 73. *Let G be a graph with precoloring (A', B') and let T be the block-cutvertex-tree of G rooted at some block node R.*

(1) A cutvertex c is forced `amber` *in $G[T_c]$, if and only if c is forced* `amber` *in $G[T_Q]$ for a child Q of c or if c is critical in both, $G[T_Q]$ and $G[T_{Q'}]$, for two different children Q and Q' of c.*

(2) A cutvertex c is forced `blue` *in $G[T_c]$, if and only if c is forced* `blue` *in $G[T_Q]$ for a child Q of c.*

(3) A cutvertex c is critical in $G[T_c]$, if and only if c is critical in $G[T_Q]$ for a child Q of c.

Proof. We first show the "if"-direction: Let c be a cutvertex of G an let Q be a child of c in T. For every (A', B')-monopolar extension (A, B) of $G[T_c]$, notice that $(A \cap V(G[T_Q]), B \cap V(G[T_Q]))$ is an (A', B')-monopolar extension of $G[T_Q]$ because $G[T_Q]$ is an induced subgraph of $G[T_c]$.

(1): If c is forced `amber` in $G[T_Q]$, then c clearly is forced `amber` in $G[T_c]$. Assume that c has another child, say Q', such that c is critical in both, $G[T_Q]$ and $G[T_{Q'}]$, that is, c has at least one `blue` neighbor in $G[T_Q]$ in every (A', B')-monopolar extension of $G[T_Q]$ and c has at least one `blue` neighbor in $G[T_{Q'}]$ in every (A', B')-monopolar extension of $G[T_{Q'}]$. Hence, c has at least two `blue` neighbors in every (A', B')-monopolar extension (A, B) of $G[T_c]$, say $w \in V(G[T_Q]) \cap B$ and $w' \in V(G[T_{Q'}]) \cap B$. This implies that c is `amber` colored in every (A', B')-monopolar extension because otherwise w–c–w' would be a `blue` colored induced P_3 in $G[T_c]$.

(2),(3): Clearly, if c is forced `blue` or critical in $G[T_Q]$, then c is also forced `blue` or critical in $G[T_c]$ respectively.

Next, we show the "only if"-direction:

(1): Let c be forced `amber` in $G[T_c]$. Assume that c is not forced `amber` in $G[T_Q]$ for every child Q of c. If c is critical in at most one of its children, then combining (A', B')-monopolar extensions of the children of c that are chosen such that c has no `blue` colored neighbor, if possible, clearly yields an (A', B')-monopolar extension of $G[T_c]$ that colors c `blue`—this is a contradiction. Hence, if x if forced `amber` in $G[T_c]$, then it is is forced `amber` in $G[T_Q]$ for a child Q of c or c is critical in $G[T_Q]$ and $G[T_{Q'}]$ for two different children Q and Q' of c.

(2): Let c be forced `blue` in $G[T_c]$. Assume that c is not forced `blue` in $G[T_Q]$ for every child Q of c. Combining (A', B')-monopolar extensions of the children of c that are chosen such that c is colored `amber` clearly yields an (A', B')-monopolar extension of $G[T_c]$ that colors c `amber`—this is a contradiction.

(3): Let c be critical in $G[T_c]$. Assume that c is not critical in $G[T_Q]$ for every child Q of c. Combining (A', B')-monopolar extensions of the children of c that are chosen such that c has no `blue` colored neighbor clearly yields an (A', B')-monopolar extension of $G[T_c]$ in which c has only `amber` colored neighbors—this is a contradiction. □

Finally, we need a way to solve MONOPOLAR EXTENSION on $G[T_Q]$ for a block Q of G while taking the information about forced `amber`, forced `blue`, and critical children into account:

Lemma 74. *Let G be a locally A_5-$S_{2,2,2}$-defused graph with precoloring (A', B'), let T be the block-cutvertex-tree of G rooted at some block node R, and let Q be a block of G. Given the sets $W_{\mathtt{a}}$, $W_{\mathtt{b}}$, and W_c containing the forced* `amber`*, forced* `blue`*, and critical children of Q in the corresponding subtree of T,* MONOPOLAR EXTENSION *on $G[T_Q]$ can be reduced to* MONOPOLAR EXTENSION *on a A_5-$S_{2,2,2}$-defused graph $Q_{\mathtt{abc}}$ of size $|V(Q_{\mathtt{abc}})| \leq 2|V(Q)|$.*

Proof. If $W_{\mathtt{a}} \cap W_{\mathtt{b}} \neq \emptyset$, there is a child c of Q that is simultaneously forced `amber` and forced `blue` in $G[T_c]$ and hence, $G[T_c]$ is not (A', B')-monopolar extendable. Since $G[T_c]$ is an induced subgraph of $G[T_Q]$, this implies that $G[T_Q]$ is not (A', B')-monopolar extendable, too. In that case, let $Q_{\mathtt{abc}}$ simply be any graph without an (A', B')-monopolar extension.

Hence, we can assume that $W_{\mathtt{a}}$ and $W_{\mathtt{b}}$ are disjoint, that is, for every child c of Q, $G[T_c]$ is (A', B')-monopolar extendable.

Starting from Q, construct $Q_{\mathtt{abc}}$ by adding a vertex c' and the edge cc' for each $c \in W_c$. Let W'_c be the set containing the added vertices,

$$\begin{aligned} A'_{\mathtt{abc}} &:= (A' \cap Q) \cup W_{\mathtt{a}}, \text{ and} \\ B'_{\mathtt{abc}} &:= (B' \cap Q) \cup W_{\mathtt{b}} \cup W'_c. \end{aligned}$$

Since we only added `blue` precolored pending vertices to Q, by Lemma 71, $Q_{\mathtt{abc}}$ is A_5-$S_{2,2,2}$-defused.

We show: $G[T_Q]$ is (A', B')-monopolar extendable, if and only if $Q_{\mathtt{abc}}$ is $(A'_{\mathtt{abc}}, B'_{\mathtt{abc}})$-monopolar extendable.

Let (A, B) be an (A', B')-monopolar extension of $G[T_Q]$. Let $A_{\mathtt{abc}} := A \cap V(Q)$ and $B_{\mathtt{abc}} := (B \cap V(Q)) \cup W'_c$. Since $A_{\mathtt{abc}}$ clearly is independent in Q and Q is an induced subgraph of $G_{\mathtt{abc}}$, $A_{\mathtt{abc}}$ is also independent in $Q_{\mathtt{abc}}$. If $Q_{\mathtt{abc}}[B_{\mathtt{abc}}]$ contains an induced P_3, it has the form c'–c–z for some $c' \in W'_c$, $c \in W_c$, and $z \in V(Q)$. But since c is critical in $G[T_c]$, there is a neighbor $u \in V(G[T_c])$ of c with $u \in B$ and, hence, u–c–z is induced in $G[B]$—this is a contradiction.

Since, for every child c of Q, $G[T_c]$ is an induced subgraph of $G[T_Q]$, by definition of $W_{\mathtt{a}}$ and $W_{\mathtt{b}}$, it must be $W_{\mathtt{a}} \subseteq A$ and $W_{\mathtt{b}} \subseteq B$. Hence, $(A_{\mathtt{abc}}, B_{\mathtt{abc}})$ is an $(A'_{\mathtt{abc}}, B'_{\mathtt{abc}})$-monopolar extension of $Q_{\mathtt{abc}}$.

Conversely, let $(A_{\mathtt{abc}}, B_{\mathtt{abc}})$ be an $(A'_{\mathtt{abc}}, B'_{\mathtt{abc}})$-monopolar extension of $Q_{\mathtt{abc}}$. For every child c of Q, let (A_c, B_c) be an (A', B')-monopolar extension of $G[T_c]$. Such a monopolar extension exists because $c \notin W_{\mathtt{a}} \cap W_{\mathtt{b}}$. If $c \in B_{\mathtt{abc}}$ and $c \notin W_c$, that is, c is not critical in $G[T_c]$, we can choose (A_c, B_c) such that $N(c) \subseteq A_c$. Let A contain the union of $A_{\mathtt{abc}}$ and A_c, for every child c of Q, and let B contain the union of $(B_{\mathtt{abc}} \setminus W'_c)$ and B_c, for every child c

of Q. Obviously, A is an independent set in $G[T_Q]$ because every edge of $G[T_Q]$ is either in Q or in $G[T_c]$ for some child c of Q. If $G[T_Q][B]$ contains an induced P_3, it has the form $c''-c-z$ for a child c of Q and vertices $c'' \in G[T_c]$ and $z \in Q$. But then, by construction of (A_c, B_c), c is critical in $G[T_c]$ and $Q_{\mathtt{abc}}[B_{\mathtt{abc}}]$ contains the induced P_3 $c'-c-z$ with c' being the vertex of W'_c adjacent to c—this is a contradiction. Hence, (A, B) is an (A', B')-monopolar extension of $G[T_Q]$.

By construction, $Q_{\mathtt{abc}}$ has $|W_c| \leq |V(Q)|$ more vertices than Q, hence, $|V(Q_{\mathtt{abc}})| \leq 2|V(Q)|$. □

It remains to show that Lemma 73 and Lemma 74 can be combined to a recursion scheme:

Lemma 75. *Let G be a locally A_5-$S_{2,2,2}$-defused graph with precoloring (A', B'), let T be the block-cutvertex-tree rooted at some block node R, let c be a cutvertex of G, and let $\mathcal{Q}$ be the set of all blocks of $G[T_c]$. Then it can be decided*

- *if c is forced* `amber` *in $G[T_v]$,*
- *if c is forced* `blue` *in $G[T_v]$, and*
- *if c is critical in $G[T_v]$*

by solving three MONOPOLAR EXTENSION *instances on an A_5-$S_{2,2,2}$-defused graph $Q_{\mathtt{abc}}$ for every $Q \in \mathcal{Q}$, whereby $|V(Q_{\mathtt{abc}})| \leq 2|V(Q)|$.*

Proof. We will show this by induction on the structure of T.

If c has no grandchildren, we have $G[T_Q] = Q$ for every child Q of c. Since the blocks of G are A_5-$S_{2,2,2}$-defused, the lemma follows from Observation 72 and Lemma 73.

Otherwise, let $c_1, \ldots, c_\ell$ be the grandchildren of c. For every grandchild c_i of c, let $\mathcal{Q}_i$ contain the blocks of the graph $G[T_{c_i}]$. By induction, we can decide whether a grandchild c_i is forced `amber`, forced `blue`, or critical by solving three MONOPOLAR EXTENSION instances on an A_5-$S_{2,2,2}$-defused graph $Q'_{\mathtt{abc}}$ for every $Q' \in \mathcal{Q}_i$, whereby $|V(Q'_{\mathtt{abc}})| \leq 2|V(Q')|$. Hence, we can compute the sets $W_{\mathtt{a}}$, $W_{\mathtt{b}}$, and W_c for all children of c by solving three MONOPOLAR EXTENSION instances on an A_5-$S_{2,2,2}$-defused graph $Q'_{\mathtt{abc}}$ for every $Q' \in (\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_\ell)$, whereby $|V(Q'_{\mathtt{abc}})| \leq 2|V(Q')|$. Having these sets computed, we can apply Lemma 74, that is, for every child Q of c, we can solve MONOPOLAR EXTENSION on $G[T_Q]$ by solving MONOPOLAR EXTENSION on an A_5-$S_{2,2,2}$-defused graph $Q_{\mathtt{abc}}$ with $|V(Q_{\mathtt{abc}})| \leq 2|V(Q)|$. By Observation 72, for a child Q of c, we can check if c is forced `amber`, forced `blue`, or critical in $G[Q]$ by solving three MONOPOLAR EXTENSION instances on $G[Q_{\mathtt{abc}}]$. With this information, Lemma 73 allows us to easily decide whether c is forced `amber`, forced `blue`, or critical in $G[T_c]$. Since $\mathcal{Q}$ is the union of $\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_\ell$ and the children of c, the lemma follows. □

This enables us to prove:

Theorem 15. MONOPOLAR EXTENSION *and, hence,* MONOPOLARITY *is solvable in time $O(n^4)$ on locally A_5-$S_{2,2,2}$-defused graphs, where n is the number of vertices of the input graph.*

Proof. Let G be a locally A_5-$S_{2,2,2}$-defused graph with precoloring (A', B'). The block-cutvertex-tree T of G can be computed in linear time, for example as a byproduct of the well-known algorithm of Hopcroft and Tarjan [56] that finds all biconnected components of a graph. Let T be rooted at some block node R. Let $\mathcal{Q}$ be the set of all blocks of G. By definition, G is (A', B')-monopolar extendable, if and only if $G[T_R]$ is (A', B')-monopolar extendable. By Lemma 74, the question if $G[T_R]$ is (A', B')-monopolar extendable can be answered by determining the sets of forced `amber`, forced `blue`, and critical children of R in T and solving MONOPOLAR EXTENSION on an A_5-$S_{2,2,2}$-defused graph $R_{\mathtt{abc}}$ with $|V(R_{\mathtt{abc}})| \leq 2|V(R)|$. By Lemma 75, finding the sets of forced `amber`, forced `blue`, and critical children of R can be done by solving three MONOPOLAR EXTENSION instances on an A_5-$S_{2,2,2}$-defused graph $Q_{\mathtt{abc}}$ for every $Q \in (\mathcal{Q} \setminus \{R\})$, where $|V(Q_{\mathtt{abc}})| \leq 2|V(Q)|$. Hence, to solve MONOPOLAR EXTENSION on G, by Theorem 14, we need time $\sum_{Q \in \mathcal{Q}} O(|Q_{\mathtt{abc}}|^4)$. Since, by Fact 1, $\sum_{Q \in \mathcal{Q}} |V(Q)| \leq 2|V(G)|$ which implies $\sum_{Q \in \mathcal{Q}} |V(Q_{\mathtt{abc}})| \leq 4|V(G)|$ because $|V(Q_{\mathtt{abc}})| \leq 2|V(Q)|$ for all $Q \in \mathcal{Q}$, we can solve MONOPOLAR EXTENSION on G in time $O(|V(G)|^4)$. □

Corollary 76. MONOPOLAR EXTENSION *and, hence,* MONOPOLARITY *is solvable in time $O(n^4)$ on hole-free graphs, where n is the number of vertices of the input graph.*

Proof. Clearly, hole-free graphs are A_5-free. The blocks of hole-free graphs are $S_{2,2,2}$-free because, if a biconnected graph contains an induced $S_{2,2,2}$, then, since all pairs of non-adjacent vertices are connected by at least two vertex-disjoint paths, the graph contains an induced cycle of length at least 5. Hence, hole-free graphs are locally $\{A_5, S_{2,2,2}\}$-free and the corollary follows by Theorem 15. □

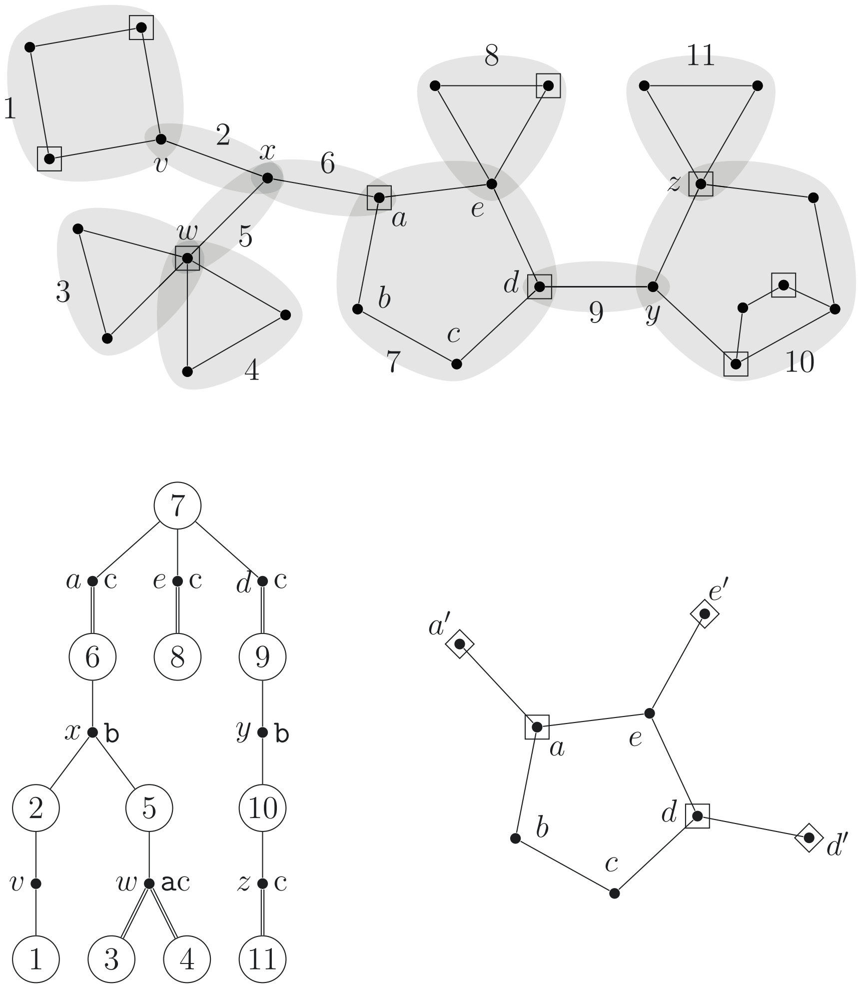

Figure 4.5: An example graph G with its 11 blocks shaded in gray (top), the block-cutvertex-tree T of G rooted at block 7 (bottom left), and the graph $7_{\mathtt{ab}c}$ (bottom right), which is the MONOPOLAR EXTENSION instance of the final step of the recursion. Coloring the vertices of G that are drawn in a square `amber` and all other vertices `blue` yields a monopolar partition of G. In T, if a cutvertex is critical in a block, then the appropriate edge is doubly drawn. If a cutvertex is forced `amber`, forced `blue`, or critical in its subtree, it is labeled with `a`, `b`, or c in T respectively. In $7_{\mathtt{ab}c}$, the precolored `blue` vertices are drawn in a diamond.

We conclude this section with an example. Consider the graph shown in the top of Figure 4.5. First notice that the graph is locally A_5-$S_{2,2,2}$-defused. We try to find a monopolar partition of this graph using the presented recursion scheme on the block-cutvertex-tree T of G, which is depicted in the bottom left of Figure 4.5. The recursion descents to the leaves of T, namely the blocks 1, 3, 4, and 11, to check if the cutvertices v, w, and z are forced `amber`, forced `blue`, or critical in these blocks.

One can easily check that block 1 admits exactly two monopolar partitions, one colors v `amber`, the other colors v `blue` and its neighbors `amber`. Hence, it is neither forced `amber`, forced `blue`, nor critical. The blocks 3, 4, and 11 are triangles and triangles have exactly three monopolar partitions, each coloring one vertex `amber` and its two neighbors `blue`. Hence, w and z are critical in these blocks, depicted by double lines for the corresponding edges of T in the figure.

By Lemma 73, w is forced `amber` because it is critical in two children and w and c are critical because they are critical in at least on child. In the figure, this is denoted by the labels `a`c and c respectively.

Next, by Lemma 74, the graphs $2_{\mathtt{ab}c}$, $5_{\mathtt{ab}c}$, and $10_{\mathtt{ab}c}$ and appropriate precolorings are constructed. Together with Observation 72, three MONOPOLAR EXTENSION instances are solved on these graphs to decide whether the cutvertices x and y are forced `amber`, forced `blue`, or critical in T_x and T_y respectively. It can easily be seen that x is forced `blue` because its neighbor w in block 5 is forced `amber`. Furthermore, y is forced `blue` because it is colored `blue` in every monopolar extension of $10_{\mathtt{ab}c}$. We omit the specification of $10_{\mathtt{ab}c}$, but we give an example for $7_{\mathtt{ab}c}$ later.

In the same way, we figure out that a, d and e are critical in T_6, T_9, and T_8 respectively.

The graph $7_{\mathtt{ab}c}$ is depicted in the bottom right of Figure 4.5. Since a, d, and e are critical in at least one child, they are critical children of 7 and the construction of $7_{\mathtt{ab}c}$ adds a pending precolored `blue` vertex to each of them. In the figure, these vertices are called a', d', and e' and the `blue` precoloring is depicted by drawing the vertices in a diamond. One can easily check that coloring the vertices in squares `amber` and all other vertices `blue` is the only monopolar extension of $7_{\mathtt{ab}c}$.

Since 7 is the root of G, this shows that G is monopolar.

4.4 Polarity on Subclasses of Planar Graphs

POLARITY is known to be NP-complete on planar graphs as we show in Section 5.4. This section gives a framework for solving POLARITY on subclasses of planar graphs. The framework requires to solve four problems as subroutines, which are UNIPOLARITY, MONOPOLARITY, and two special cases of MONOPOLAR EXTENSION. We show that one of the special cases of MONOPOLAR EXTENSION is efficiently solvable on every planar graph using the result of Section 4.2. Since UNIPOLARITY is efficiently solvable on every graph but POLARITY is NP-complete on planar graphs, at least one of the two other problems is NP-complete on every subclass on which POLARITY remains NP-complete. Nonetheless, we identify subclasses of planar graphs, for example hole-free planar graphs and maximal planar graphs, for which the complexity of POLARITY was open, whereas the framework provides an efficient algorithm. In fact, we show that POLARITY is polynomial-time solvable on every hereditary subclass of planar graphs that admits efficient MONOPOLAR EXTENSION.

The input graphs for the framework do not need to be planar, it suffices that they are K_5-free and do not contain $K_{3,3}$ as a subgraph. Since planar graphs are $\{K_5, K_{3,3}\}$-minor free, every planar graph fulfills this condition. The framework is based on the following observation:

Observation 77. *Let G be a K_5-free graph without $K_{3,3}$ as a subgraph, let (A, B) be a polar partition of G, and let $A_1, \dots, A_k$ be the maximal independent sets of $G[A]$, assuming that $1 \leq |A_1| \leq \cdots \leq |A_k|$. If $|A| \geq 7$, then $k \leq 3$ and $|A| - |A_k| \leq 2$.*

Proof. Notice first that $k \leq 4$ because otherwise $G[A]$ would contain the K_5. Moreover, if $k = 4$, then $|A_1| = |A_2| = |A_3| = 1$ and $|A_4| \leq 2$ because otherwise $K_{3,3}$ would be a subgraph of $G[A]$; in particular, this implies $|A| \leq 5$. Hence, $|A| \geq 6$ implies that $k \leq 3$.

Assume that $|A| \geq 7$. Since $|A| = |A_1| + \cdots + |A_k| \leq k \cdot |A_k|$ and $k \leq 3$, we have $|A_k| \geq 3$. Hence, we have $|A| - |A_k| \leq 2$ because otherwise, the $K_{3,3}$ would be a subgraph of $G[A]$ with three vertices of $A \setminus A_k$ and three vertices of A_k. □

This observation implies that every polar partition (A, B) of a K_5-free graph without $K_{3,3}$ as subgraph with $|A| \geq 7$ has one of the following forms:

Type 1 $G[A]$ consists of an independent set of size $|A| - 1$ and a universal vertex x.

Algorithm: `PlanarPolarity`
Input: A K_5-free graph G without $K_{3,3}$ as a subgraph.

1. Check if G is unipolar or monopolar, if so, then `STOP:` G `is polar`.
2. For every $A \subseteq V$ with $|A| \leq 6$: Check if $(A, V \setminus A)$ is a polar partition of G, if so, then `STOP:` G `is polar`.
3. For every $x \in V$: Check if 1-POLARITY answers “yes” on G and x, if so, then `STOP:` G `is polar`.
4. For every $\{x, y\} \subseteq V$: Check if 2-POLARITY answers “yes” on G, x, and y, if so, then `STOP:` G `is polar`.
5. `STOP:` G `is not polar`.

Figure 4.6: A framework for solving POLARITY on planar graphs.

Type 2 $G[A]$ consists of an independent set of size $|A| - 2$ and two universal vertices x and y.

Type 3 $G[A]$ consists of an independent set of size $|A| - 2$ and two non-adjacent vertices x and y that are both $(A \setminus \{x, y\})$-universal.

Consequently, if we look for a polar partition of a K_5-free graph G without $K_{3,3}$ as a subgraph, then we can test if G admits a unipolar partition, a monopolar partition, or a polar partition (A, B) with $|A| \leq 6$. If no such partition exists, we can test if G admits a polar partition of Type 1, 2, or 3. This leads to the definition of the following problems:

Definition 16. 1-POLARITY *asks for a given graph G and vertex x, if G admits a polar partition (A, B) of Type 1 such that x is the universal vertex in $G[A]$.*

Definition 17. 2-POLARITY *asks for a given graph G and two vertices x and y, if G admits a polar partition (A, B) of Type 2 or Type 3 such that $\{x, y\} \subseteq A$ are the $(A \setminus \{x, y\})$-universal vertices in $G[A]$.*

There is no need to distinguish between Type 2 and Type 3 because, as we will see, 2-POLARITY is efficiently solvable on every planar graph. Hence, we can formulate the framework as shown in Figure 4.6.

The correctness of the framework obviously follows from Observation 77 and the foregoing considerations. Whether the framework can be executed efficiently depends on the complexity of MONOPOLARITY, 1-POLARITY, and 2-POLARITY because UNIPOLARITY is known to be polynomial-time solvable, the number of subsets of V on at most six vertices is bound by $O(|V|^6)$, and testing a partition for being polar can be done in time at most $O(|V|^3)$ by checking $G[A]$ for being $(P_2 + P_1)$-free and $G[B]$ for being P_3-free. As already mentioned, we show that 2-POLARITY is polynomial-time solvable on every planar graph, in particular on every graph without $K_{3,3}$ as subgraph:

Lemma 78. 2-POLARITY *can be solved in time* $O(|V|^4)$ *on every graph* $G = (V, E)$ *without* $K_{3,3}$ *as a subgraph.*

Proof. Let $G = (V, E)$ be a graph without $K_{3,3}$ as subgraph and let $\{x, y\} \subseteq V$. Let $C := N(\{x, y\})$. First, notice that $G[C]$ has maximum degree at most 2 because otherwise, for a vertex $c \in C$ of degree at least 3, x, y, c, and three neighbors of c in C form a $K_{3,3}$ in G—this is a contradiction.

Let $G' = (V', E') := G - \{x, y\}$. One can easily verify that G has a polar partition (A, B) of Type 2 or 3 such that $\{x, y\} \subseteq A$ are the two $(A \setminus \{x, y\})$-universal vertices in $G[A]$, if and only if G' has a monopolar partition (A_m, B_m) with $A_m \subseteq C$, that is, if and only if G' is $(\emptyset, V' \setminus C)$-monopolar extendable. Hence, to solve 2-POLARITY on G, x, and y, we can solve MONOPOLAR EXTENSION on G' with precoloring $(\emptyset, V' \setminus C)$.

Since $V' \setminus C$ is `blue` precolored, by Observation 62, every induced P_3 P of G that is not defused is completely in $G[C]$ and has no neighbor in $V' \setminus C$, that is, $N_{G'}[V(P)] \subseteq C$. This implies that every vertex of a not defused induced P_3 of G' has degree at most 2 because $G[C]$ has maximum degree at most 2. Hence, by Definition 14, G' is A_5-$S_{2,2,2}$-defused and, by Theorem 14, MONOPOLAR EXTENSION can be solved in time $O(|V'|^4) \subseteq O(|V|^4)$ on G'. □

One can easily check that 1-POLARITY can be expressed as a MONOPOLAR EXTENSION instance in the following way:

Observation 79. *For a graph* $G = (V, E)$ *and a vertex* $x \in V$*,* G *and* x *is a "yes"-instance of* 1-POLARITY*, if and only if* $G - x$ *is* $(\emptyset, V \setminus N(x))$*-monopolar extendable.*

Thus, to execute `PlanarPolarity` efficiently on subclass $\mathcal{C}$ of planar graphs, we need efficient algorithms for MONOPOLARITY and MONOPOLAR EXTENSION on $\mathcal{C}$. Since MONOPOLARITY is a special case of MONOPOLAR EXTENSION, the computational complexity of the framework on $\mathcal{C}$ is highly related to the complexity of MONOPOLAR EXTENSION on $\mathcal{C}$.

We can immediately conclude that POLARITY is polynomial-time solvable on every hereditary subclass of planar graphs that admits an efficient MONOPOLAR EXTENSION algorithm. An interesting example for such a graph class are the planar hole-free graphs, in particular because POLARITY is NP-complete on planar graphs and on hole-free graphs, as we show in Section 5.4.

Since locally $\{A_5, S_{2,2,2}\}$-free graphs are hereditary, Theorem 15, Corollary 76, Lemma 78, Observation 79, and the framework imply:

Theorem 16. POLARITY *can be solved in polynomial time on planar locally* $\{A_5, S_{2,2,2}\}$*-free graphs and, hence, on hole-free planar graphs and chair-free planar graphs.*

Beside hereditary graph classes, we can apply the framework as long as we can show that MONOPOLARITY and 1-POLARITY are tractable. As an example, we analyze maximal planar graphs.

Lemma 80. 1-POLARITY *can be solved in polynomial time on maximal planar graphs.*

Proof. Let $G = (V, E)$ be a maximal planar graph with $|V| \geq 4$ and let $x \in V$. By the maximal planarity of G, any edge of G belongs to two different triangles in G. It follows that every edge of $G - x$ belongs to at least one triangle in $G - x$. This means that $G - x$ is P_3-defused. Since, by Observation 79, 1-POLARITY on G can be solved by MONOPOLAR EXTENSION on $G - x$ and, by Theorem 13, MONOPOLAR EXTENSION is tractable on P_3-defused graphs, the lemma follows. $\square$

Theorem 13, which states that MONOPOLARITY is polynomial-time solvable on maximal planar graphs, and Lemma 80 together imply:

Theorem 17. POLARITY *runs in polynomial time on maximal planar graphs.*

5 NP-completeness Results

This chapter presents a reduction framework for NP-completeness proofs. The framework is used to establish NP-completeness results for all of our considered problems. Sections 5.2 and 5.3 show that both, EFFICIENT DOMINATION and EFFICIENT EDGE DOMINATION, are NP-complete on planar bipartite graphs with maximum degree at most 3 and girth at least g, for every fixed g. For MONOPOLARITY and POLARITY, Section 5.4 shows that the problems remain NP-complete on triangle-free planar graphs with maximum degree at most 3 and on $\{C_4, \ldots, C_g\}$-free planar graphs with maximum degree at most 3, for every fixed $g \geq 4$. Furthermore, POLARITY is NP-complete on the complements of these two classes.

The reduction framework and the concrete results are unified versions of the NP-completeness results given in [8, 12, 13, 63, 64].

5.1 Reduction Framework

The framework describes a reduction from a 3-SATISFIABILITY variant called MONOTONE PLANAR ONE-IN-THREE 3-SAT (MPOIT 3-SAT, for short) to a graph problem. MPOIT 3-SAT is shortly introduced in Section 5.1.1.

The reduction only works for graph problems that can be formulated as the decision whether an input graph G admits a vertex subset D with a certain property Π. In particular, EFFICIENT DOMINATION, EFFICIENT EDGE DOMINATION, and MONOPOLARITY can be formulated by the following vertex subset properties:

- EFFICIENT DOMINATION: Π requires that D is efficient dominating in G.

- EFFICIENT EDGE DOMINATION: Π requires that D is independent in G and $G - D$ is 1-regular.

- MONOPOLARITY: Π requires that $(D, V(G)\backslash D)$ is a monopolar partition of G.

Furthermore, the property must be *additive*: If F and G are graphs and $D_F \subseteq V(F)$ fulfills Π in F and $D_G \subseteq V(G)$ fulfills Π in G, then $D_F \cup D_F$ must fulfill Π in $(G + F)$. One can easily check that this is true for the three mentioned problems.

For an input formula F, we construct a reduction graph $G(F)$ such that F has a satisfying truth assignment b, if and only if $G(F)$ admits a Π-fulfilling vertex subset D. In particular, $G(F)$ has a vertex v_i for every variable Y_i of F such that $b(Y_i) = \texttt{true} \Leftrightarrow v_i \in D$.

The reduction framework describes how to combine three types of gadgets to construct $G(F)$: an initial-gadget that provides the vertex v_i for every variable Y_i of F, a copy-gadget that transports the truth values in $G(F)$, and a clause-gadget that ensures that the truth values satisfy F. In Section 5.1.2, we describe these gadgets in an abstract way. The application of the framework for a concrete problem, that is, for a concrete property Π, requires the implementation of the gadgets.

We design the reduction framework so that some structural properties of the gadgets are preserved. More precisely, $G(F)$ is supposed to be bipartite, if the gadgets are bipartite, $G(F)$ is supposed to be planar, if the gadgets are planar, $G(F)$ is supposed to have maximum degree at most d ($d \geq 3$), if the gadgets have maximum degree at most d, and $G(F)$ is supposed to have girth at least g, if the gadgets have girth at least g. The reduction is described in Section 5.1.3.

5.1.1 Monotone Planar One-in-Three 3-Sat

ONE-IN-THREE 3-SAT is a variant of the well known BOOLEAN SATISFIABILITY problem (SAT for short). Given a boolean formula F in conjunctive normal form with boolean variables Y and clauses C, BOOLEAN SATISFIABILITY asks if there is a truth assignment $b : Y \to \{\texttt{true}, \texttt{false}\}$ such that every clause contains at least one true literal. SAT was the first problem shown to be NP-complete [28] and forms a starting point for several reductions showing NP-completeness of a wide variety of problems. In [58], along with 20 basic combinatorial problems, the variant 3-SAT of SAT is proved to be NP-complete. In 3-SAT, the input formula in conjunctive normal form is restricted to have exactly three literals per clause. Besides SAT and 3-SAT, [47] contains several variants of SAT that are NP-complete, including ONE-IN-THREE 3-SAT.

In ONE-IN-THREE 3-SAT, the input formula is also restricted to contain no clause with more than three literals, but we ask for a truth assignment b such that every clause contains exactly one true literal. By [61] and [79], it is known that ONE-IN-THREE 3-SAT remains NP-complete, even if the input formula has a planar incidence graph and contains no negations. The *incidence graph* of a formula F with variables Y and clauses C is

$$\mathcal{I}(G) := (Y \cup C, \{vc \mid v \in Y, c \in C, v \in c\}),$$

that is, the graph that has the variables and clauses of F as vertices and an edge between a variable vertex and a clause vertex, if and only if the corresponding variable appears in the corresponding clause in F. We say that a formula F is planar, if $\mathcal{I}(F)$ is planar. The variant of ONE-IN-THREE 3-SAT restricted to planar formulas without negation is called MONOTONE PLANAR ONE-IN-THREE 3-SAT (MPOIT 3-SAT, for short). MPOIT 3-SAT is known as a standard tool for showing NP-completeness of geometric problems, as stated in [78].

5.1.2 The Gadgets

We define gadget templates for the initial-gadget, the copy-gadget, and the clause-gadget by defining properties that the concrete gadgets must fulfill. Whenever it is possible to implement concrete gadgets for a specific graph problem, the reduction in the next section shows NP-completeness for this problem. Notice that our clause-gadget is called one-in-three-gadget because we reduce from MPOIT 3-SAT.

Definition 18 (initial-gadget). *The* initial-gadget $I(x)$ *is a graph with a particular vertex* x *that admits at least two* Π*-fulfilling vertex subsets,* D *and* D'*, such that* $x \in D$ *and* $x \notin D'$*.*

Attaching the copy-gadget to a vertex x of a graph G transports the value of x with respect to a Π-fulfilling set D of G to a new vertex y, that is, $x \in D \Leftrightarrow y \in D$.

Definition 19 (copy-gadget). *Let* G *be any graph with a particular vertex* x'*. The* copy-gadget $C(x, y)$ *is a graph with particular distinct vertices* x *and* y *such that the following holds: If* G' *is the union of* G *and* $C(x, y)$ *on* $x = x'$*, then*

(1) for every Π*-fulfilling set* D' *of* G'*, we have either* $\{x, y\} \subseteq D'$ *or* $\{x, y\} \cap D' = \emptyset$ *and*

Figure 5.1: Pictographs for the gadgets $I(x)$, $C(x, y)$, $CC^g(x, y)$ and $O(x, y, z)$ (from left to right).

(2) for every Π*-fulfilling set* D *of* G*, there is a* Π*-fulfilling set* D' *of* G' *with* $D \subseteq D'$*.*

For convenience, we also define:

Definition 20 (copy-chain-gadget)**.** *The* copy-chain-gadget $CC^g(x, y)$ *is the union of* g *copies of the copy-gadget,* $C_1(x_1, y_1), \ldots, C_g(x_g, y_g)$ *on* $y_1 = x_2$*,* $y_2 = x_3$*, …,* $y_{g-1} = x_g$*, where* x *equals* x_1 *and* y *equals* y_g*.*

Attaching the one-in-three-gadget to the vertices x, y, and z of a graph G yields a graph G' that inherits all Π-fulfilling sets of G that contain exactly one of x, y, and z.

Definition 21 (one-in-three-gadget)**.** *Let* G *be any graph with particular vertices* x'*,* y'*, and* z'*. The* one-in-three-gadget $O(x, y, z)$ *is a graph with particular distinct vertices* x*,* y*, and* z *such that the following holds: If* G' *is the union of* G *and* $O(x, y, z)$ *on* $x = x'$ *and* $y = y'$ *and* $z = z'$*, then*

(1) for every Π*-fulfilling set* D' *of* G'*, we have* $|\{x, y, z\} \cap D'| = 1$ *and*

(2) for every Π*-fulfilling set* D *of* G *with* $|\{x, y, z\} \cap D| = 1$*, there is a* Π*-fulfilling set* D' *of* G' *with* $D \subseteq D'$*.*

Figure 5.1 shows pictographs of these gadgets that are used later for better readability.

5.1.3 The Reduction Scheme

Let Π be a vertex subset property, let $I(x)$ be a concrete initial-gadget, let $C(x, y)$ be a concrete copy-gadget, and let $O(x, y, z)$ be a concrete one-in-three-gadget. Let F be a planar boolean formula in conjunctive normal form with variable set Y and clause set C without negation and with exactly three literals per clause and let $|Y| = n$ and $|C| = m$. We construct a reduction graph $G(F)$ such that F is satisfiable with exactly one true literal per clause, if and only if $G(F)$ admits a Π-fulfilling vertex subset D. Our goal is to have a vertex v_i in $G(F)$ for every variable $Y_i \in Y$ such that v_i's containment in a Π-fulfilling set D of G corresponds to the truth value of Y_i in a truth assignment of F with exactly one true literal per clause. Precisely spoken, F admits a truth assignment b with exactly one true literal per clause, if and only if $G(F)$ admits a Π-fulfilling set D such that $b(Y_i) = \texttt{true}$, if $v_i \in D$, and $b(Y_i) = \texttt{false}$, if $v_i \notin D$.

For better comprehension, Figure 5.2 shows an example of the reduction for a specific formula.

The vertices $v_1, \ldots, v_n$ are established with the initial-gadget. Hence, we start by putting disjoint copies of the initial-gadget $I(v_i)$ for every variable $Y_i \in Y$ into a graph $G_1(F)$.

Ideally, we would like to attach a one-in-three-gadget for every clause of F to the corresponding vertices of the initial gadgets in $G_1(F)$. But this would imply that the degree of a vertex v_i can increase up to m. To fulfill the intended maximum degree constraints for $G(F)$, we copy the value of v_i to m vertices, namely $v'_{i,1}, \ldots, v'_{i,m}$. With this, we have a copy of the value of v_i for every clause of F. We create these copies using the copy-gadget by defining $G_2(F)$ inductively: Let $G_{2,0,0}(F) := G_1(F)$, where v_i is renamed to $v'_{i,1}$. For every $i \in \{1, \ldots, n\}$, the graph $G_{2,i,0}(F)$ is defined as the graph $G_{2,i-1,m}(F)$. For every $j \in \{1, \ldots, m\}$, the graph $G_{2,i,j}(F)$ is defined as the union of $G_{2,i,j-1}(F)$ and two copies of the copy-gadget, $C(x', v''_{i,j})$ and $C(x'', v'_{i,j})$, on $x' = v'_{i,j-1}$ and on $x'' = v''_{i,j}$. Finally, $G_2(F)$ is defined as the graph $G_{2,n,m}(F)$. That is, we produce m copies of every v_i by a chain of copy-gadgets. Taking two copies of the copy-gadget ensures that the distance between $v'_{i,j}$ and $v'_{i,k}$ is even for every $1 \leq i \leq n$ and every $1 \leq j, k \leq m$. This makes is possible to show that $G(F)$ is bipartite, if the gadgets are bipartite. In Figure 5.2, $G_2(F)$ corresponds to the parts that are inside the dashed circles.

For preserving the girth of the gadgets, we generate further copies of v_i that pairwise have a distance of at least g for some fixed g. We create these copies using the copy-chain-gadget by defining $G_3(F)$ inductively: Let $G_{3,0,0}(F) := G_2(F)$ and, for every $i \in \{1, \ldots, n\}$, let $G_{3,i,0}(F)$ be defined as the graph $G_{3,i-1,m}(F)$. For every $j \in \{1, \ldots, m\}$, the graph $G_{3,i,j}(F)$ is defined as the union of $G_{3,i,j-1}(F)$ and a copy of the copy-chain-gadget $CC^g(x, v_{i,j})$ on $x = v'_{i,j}$. Finally, $G_3(F)$ is defined as the graph $G_{3,n,m}(F)$.

We connect the copies of the variables using the one-in-three-gadget according to the clauses $C_1, \ldots, C_m$ of F. For this, we define $G(F)$ inductively: Let $G_{4,0}(F) := G_3(F)$. For every clause $C_j = \{Y_r, Y_s, Y_t\}$, $G_{4,j}(F)$ is the union of $G_{4,j-1}(F)$ and a copy of the one-in-three-gadget $O(x, y, z)$ on $x = v_{r,j}$ and

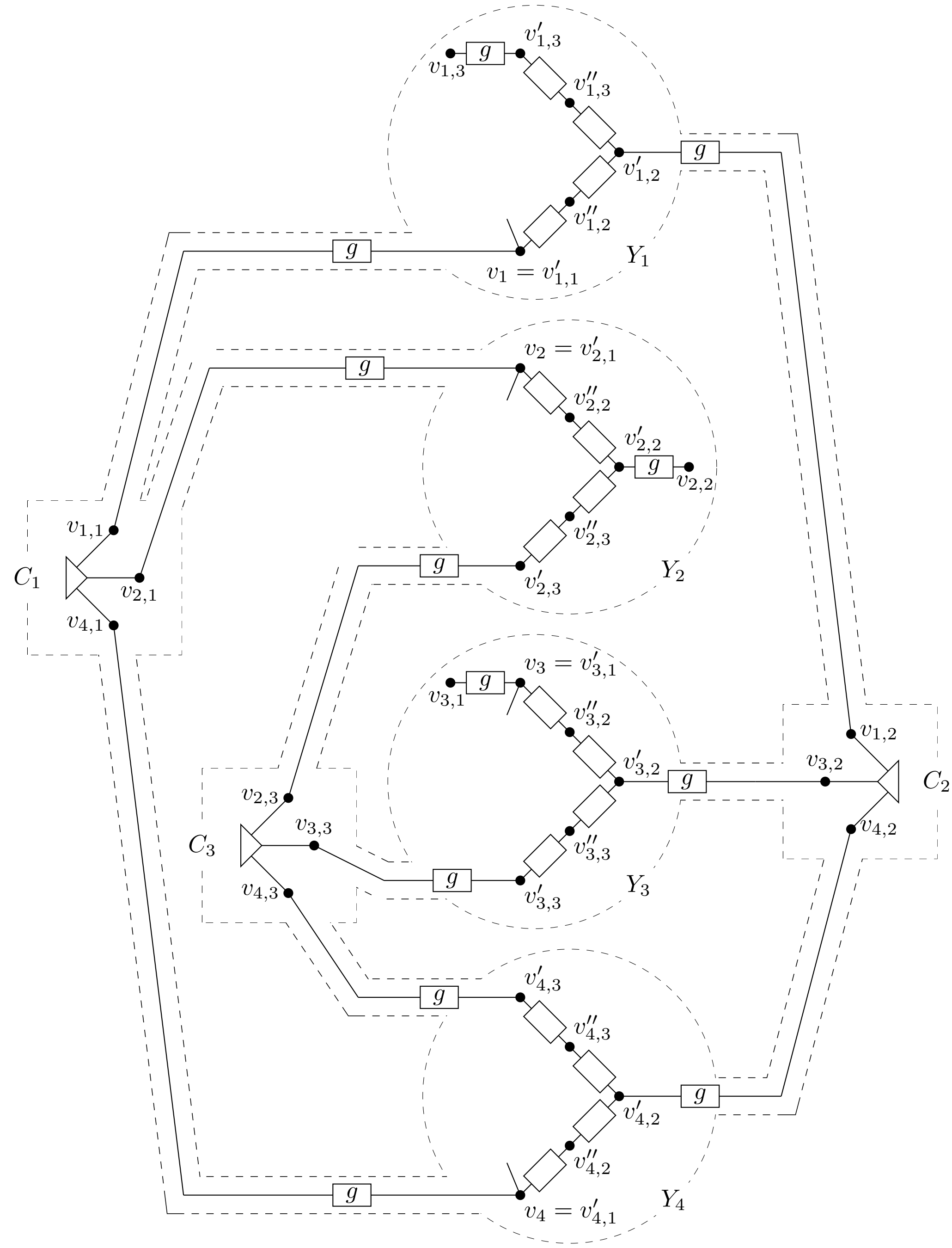

Figure 5.2: The graph $G(F)$ for $F = (C_1 \wedge C_2 \wedge C_3)$ with $C_1 = (Y_1 \vee Y_2 \vee Y_4)$, $C_2 = (Y_1 \vee Y_3 \vee Y_4)$, and $C_3 = (Y_2 \vee Y_3 \vee Y_4)$. The dashed shape depicts the planar incidence graph $I(F)$ with round variable vertices and rectangular clause vertices. Obviously, $G(F)$ follows the structure of $I(F)$.

$y = v_{s,j}$ and $z = v_{t,j}$. At the end, the reduction graph $G(F)$ is defined as $G_{4,m}(F)$. In Figure 5.2, $G_3(F)$ introduces the edges in the dashed shape and $G_4(F)$ attaches the clause-gadgets, that are depicted in dashed boxes.

Lemma 81. *Let F be a boolean formula in conjunctive normal form without negation and exactly three literals per clause. There is a truth assignment of F that values exactly one literal per clause* `true`*, if and only if the reduction graph $G(F)$ admits a Π-fulfilling set D.*

Proof. Let Y be the set of variables and C be the set of clauses of F and let $|Y| = n$ and $|C| = m$.

For the first direction, let $b : Y \to \{\texttt{true}, \texttt{false}\}$ be a truth assignment of F that values exactly one literal per clause `true`. For every copy of the initial-gadget in $G_1(F)$, let D_1^i be a Π-fulfilling set with $v_i \in D_1^i \Leftrightarrow b(Y_i) = \texttt{true}$. These sets exist by Definition 18. By Definition 19, we know that we can repeatedly extend D_1^i to D_2^i for every $i \in \{1, \dots, n\}$ such that D_2, which is the union of all D_2^i, is a Π-fulfilling set of $G_2(F)$ with

$$\begin{aligned} v''_{i,1} \in D_2 \Leftrightarrow v_i \in D_1 \ &, \ v'_{i,1} \in D_2 \Leftrightarrow v_i \in D_1 \ , \\ v''_{i,2} \in D_2 \Leftrightarrow v_i \in D_1 \ &, \ v'_{i,2} \in D_2 \Leftrightarrow v_i \in D_1 \ , \\ &\dots , \\ v''_{i,m} \in D_2 \Leftrightarrow v_i \in D_1 \ &, \ v'_{i,m} \in D_2 \Leftrightarrow v_i \in D_1 . \end{aligned} \tag{5.1}$$

Analogously, by Definitions 19 and 20, we know that we can repeatedly extend D_2^i to D_3^i for every $i \in \{1, \dots, n\}$ such that D_3, which is the union of all D_3^i, is a Π-fulfilling set of $G_3(F)$ with

$$v_{i,1} \in D_3 \Leftrightarrow v'_{i,1} \in D_2 \ , \ v_{i,2} \in D_3 \Leftrightarrow v'_{i,2} \in D_2 \ , \ \dots \ , \ v_{i,m} \in D_3 \Leftrightarrow v'_{i,m} \in D_2,$$

and, by (5.1),

$$v_{i,1} \in D_3 \Leftrightarrow v_i \in D_1 \ , \ v_{i,2} \in D_3 \Leftrightarrow v_i \in D_1 \ , \ \dots \ , \ v_{i,m} \in D_3 \Leftrightarrow v_i \in D_1 .$$

Finally, Definition 21 guarantees that we can extend D_3 to a Π-fulfilling set D_4 of $G(F)$ because we know that for every clause $C_j = \{Y_r, Y_s, Y_t\}$ exactly one variable is valued `true` by b, and hence, exactly one vertex of $v_{r,j}$, $v_{s,j}$, and $v_{t,j}$ is in D_3 for every $j \in \{1, \dots, m\}$. This shows that $G(F)$ admits a Π-fulfilling set.

For the other direction, let D be a Π-fulfilling vertex subset of $G(F)$. Let $C_j = \{Y_r, Y_s, Y_t\} \in C$ be an arbitrary clause of F. Definition 21 guarantees that exactly one of the vertices $v_{r,j}$, $v_{s,j}$, and $v_{t,j}$ is in D. Hence, by Definition 20, we know that exactly one of the vertices $v'_{r,j}$, $v'_{s,j}$, and $v'_{t,j}$ is in D. By Definition 19, this also holds for the vertices v_r, v_s, and v_t. Thus, the truth assignment b with $b(Y_i) = \texttt{true} \Leftrightarrow v_i \in D$ values exactly one literal of every clause of F `true`. □

Since the construction clearly can be done in polynomial time and MPOIT 3-SAT is NP-complete, Lemma 81 immediately implies:

Theorem 18. *For an additive vertex subset property* Π *that admits the implementation of an initial-gadget, a copy-gadget, and a one-in-three-gadget, determining if a given graph* G *admits a* Π*-fulfilling set is NP-complete.*

For showing NP-completeness on restricted graph classes, it is interesting to analyze some structural properties of $G(F)$.

Lemma 82. *If* F *is a planar boolean formula in conjunctive normal form without negation and at most three literals per clause, all gadgets are planar and admit embeddings such that the vertices* x, y *and* z *are incident to the outer face, then* $G(F)$ *is planar.*

Proof. Let $I(F) = (V_F, E_F)$ be the incidence graph of F. When considering $G(F)$, we can say that $G(F)$ results from $I(F)$ by replacing the variable vertices $Y_1, \ldots, Y_n$ of $I(F)$ by the connected components of $G_2(F)$, replacing the edges of $I(F)$ by the copy-chain-gadgets that are added in $G_3(F)$ and replacing the clause vertices $C_1, \ldots, C_j$ by the one-in-three-gadgets added in $G_4(F)$. Since $I(F)$ is planar, we can use a planar embedding of $I(F)$ to construct a planar embedding of $G(F)$:

Since all gadgets are planar and x and y can be managed to touch the outer face of an embedding of $C(x, y)$, it is easy to check that replacing an edge of $I(F)$ by a copy of the copy-chain-gadget introduced no edge-crossings.

This is also the case for replacing the clause vertices of $I(F)$ with copies of the one-in-three-gadget because $O(x, y, z)$ allows an embedding with x, y, and z incident to the outer face and this embedding can be rotated, translated, scaled, and flipped to introduce no edge crossings in $G(F)$.

The most interesting part is the replacement of the variable vertices of $I(F)$ because the variable vertices can have up to m neighbors in $I(F)$. For a variable Y_i, let $C_{j_1}, \ldots, C_{j_\ell}$ with $\{j_1, \ldots, j_\ell\} \subseteq \{1, \ldots, m\}$ be the neighbors of Y_i, ordered clockwise around Y_i in the planar embedding of $I(F)$. Consider the connected components of $G_2(F)$. Each component is a copy of the initial-gadget attached to a chain of copies of the copy-gadget. Since the initial-gadget and the copy-gadget are planar with x, respectively x and y, embeddable incident to the outer face, we can choose an embedding of every connected component that bends the chain of copy-gadgets like a circle such that the vertices $v'_{\cdot,\cdot}$ lie on the outer face. We can rename $v'_{i,1}$ to v'_{i,j_1}, $v'_{i,2}$ to v'_{i,j_2}, …, $v'_{i,\ell}$ to v'_{i,j_ℓ} for every $i \in \{1, \dots, n\}$ without modifying the semantics of the reduction. Hence, we can attach every neighbor C_{j_k} of Y_i in $I(F)$ to the appropriate vertex v'_{i,j_k} using the copy-chain-gadget without introducing edge crossings. □

Lemma 83. *If all gadgets are bipartite and the distance between x, y, and z is pairwise even in the one-in-three-gadget, then $G(F)$ is bipartite.*

Proof. We show that the reduction introduces no cycles of odd length. Since all gadgets are bipartite, they clearly contain no cycles of odd length. In $G_3(F)$, every cycle is contained in any of the gadgets because the gadgets are composed in a tree. Hence, $G_3(F)$ is bipartite. Moreover, all paths between $v_{i,j}$ and $v_{i,j'}$ have even length for all $i \in \{1, \dots, n\}$ and all $j, j' \in \{1, \dots, m\}$: Without loss of generality assume that $j \leq j'$. Let $P_C(v'_{i,k}, v''_{i,k})$ be an arbitrarily chosen path from $v'_{i,k}$ to $v''_{i,k}$ for all $k \in \{j, \dots, j'\}$ and let $P_C(v''_{i,\ell}, v'_{i,\ell+1})$ be an arbitrarily chosen path from $v''_{i,\ell}$ to $v'_{i,\ell+1}$ for all $\ell \in \{j, \dots, j'-1\}$ in the appropriate copies of the copy-gadget. Furthermore, let $P_{CC}(v_{i,j}, v'_{i,j})$ be an arbitrarily chosen path from $v_{i,j}$ to $v'_{i,j}$ and let $P_{CC}(v_{i,j'}, v'_{i,j'})$ be an arbitrarily chosen path from $v_{i,j'}$ to $v'_{i,j'}$ through the appropriate copies of the copy-chain-gadget. Notice that either all paths in $C(x, y)$ from x to y have even length or all these paths have odd length because otherwise the copy-gadget is not bipartite. This implies that either all paths from x to y in the copy-chain-gadget $CC^g(x, y)$ are of even length or all these paths are of odd length. Hence, the path $P_C(v'_{i,k}, v'_{i,k+1}) := P_C(v'_{i,k}, v''_{i,k}).P_C(v''_{i,k}, v'_{i,k+1})$ has even length for every $k \in \{j, \dots, j'-1\}$. Every path between $v_{i,j}$ and $v_{i,j'}$ has the form

$$P_{CC}(v_{i,j}, v'_{i,j}).P_C(v'_{i,j}, v'_{i,j+1}).\cdots.P_C(v'_{i,j'-1}, v'_{i,j'}).P_{CC}(v'_{i,j'}, v_{i,j'})$$

and, since the sub-paths were chosen arbitrarily, all paths between $v_{i,j}$ and $v_{i,j'}$ in $G_3(F)$ have even length.

In G_4, let $P^j_C(i, i')$ be an arbitrarily chosen path from $v_{i,j}$ to $v_{i',j}$ in the copy of the one-in-three-gadget that corresponds to clause C_j and let $P^i_Y(j, j')$ be an arbitrarily chosen path from $v_{i,j}$ to $v_{i,j'}$. Every cycle that is not entirely contained in a copy of one of the gadgets has the form

$$v_{i_1,j_1} - P^{j_1}_C(i_1, i_2) - v_{i_2,j_1} - P^{i_2}_Y(j_1, j_2) - v_{i_2,j_2} - \ldots - v_{i_k,j_k} - \ldots - v_{i_{k+1},j_{k+1}},$$

where $v_{i_{k+1},j_{k+1}}$ equals v_{i_1,j_1}. We already know that the path $P_Y^i(j_\ell, j_{\ell+1})$ has even length for all $i \in \{i_1, \dots, i_k\}$ and all $\ell \in \{1, \dots, k\}$. By assumption, the distance between $v_{i,j}$ and $v_{i',j}$ in the one-in-three-gadget is even and, since the one-in-three-gadget is bipartite, also the path $P_C^j(i_\ell, i_{\ell+1})$ has even length for all $j \in \{j_1, \dots, j_k\}$ and all $\ell \in \{1, \dots, k\}$. Hence, the length of the cycle is also even. □

Lemma 84. *Let m_g be the maximum degree of the initial-gadget, the copy-gadget, and the one-in-three-gadget. Let m_i be the degree of x in $I(x)$, let m_c be the maximum degree of x and y in $C(x,y)$, and let m_o be the maximum degree of x, y, and z in $O(x,y,z)$. The maximum degree of $G(F)$ is* $\max\{m_g, m_i + m_c, 3m_c, m_c + m_o\}$.

Proof. Obviously, the maximum degree of $G(F)$ is the maximum of the maximum degree inside the gadgets, the degree of the vertices $v_1, \dots, v_n$ that connect the initial-gadget and the copy-gadget, the degree of the vertices $v'_{i,j}$ for all $i \in \{1, \dots, n\}$ and all $j \in \{1, \dots, m\}$ that connect three copies of the copy-gadget, and the degree of the vertices $v_{i,j}$ for all $i \in \{1, \dots, n\}$ and all $j \in \{1, \dots, m\}$ that connect the copy-gadget and the one-in-three-gadget. □

Lemma 85. *If the initial-gadget, the copy-gadget and the one-in-three-gadget are $\{C_k, \dots, C_{g-1}\}$-free, then $G(F)$ is $\{C_k, \dots, C_{g-1}\}$-free. In particular, if the initial-gadget, the copy-gadget and the one-in-three-gadget have girth at least g, then $G(F)$ has girth at least g.*

Proof. If all gadgets are $\{C_k, \dots, C_{g-1}\}$-free, then clearly every induced cycle of $G(F)$ that is part of a copy of one of the gadgets has length ℓ with $\ell < k$ or $\ell \geq g$. By construction of $G(F)$, every induced cycle C that is not part of a copy of one of the gadgets passes at least two copies of the copy-chain-gadget. Since in the copy-gadget $C(x,y)$, the distance between x and y is at least 1, every path from x' to y' in a copy of the copy-chain-gadget $CC^g(x', y')$ has length at least g. This immediately implies that C has length at least $2g$. Hence, every induced cycle of $G(F)$ as length at most $k-1$ or at least g. □

We can use these structural relations with the following corollary of Theorem 18:

Corollary 86. *For an additive vertex subset property Π that admits the implementation of an initial-gadget, a copy-gadget, and a one-in-three-gadget, deciding if an input graph G admits a Π-fulfilling set is NP-complete. If the gadgets can be implemented such that*

- *$G(F)$ is planar, then this decision remains NP-complete on planar graphs.*
- *$G(F)$ is bipartite, then this decision remains NP-complete on bipartite graphs.*
- *$G(F)$ has maximum degree at most 3, then this decision remains NP-complete on graphs with maximum degree at most 3.*
- *$G(F)$ is $\{C_k, \ldots, C_{g-1}\}$-free, then this decision remains NP-complete on $\{C_k, \ldots, C_{g-1}\}$-free graphs.*

5.2 Efficient Domination

We can formulate EFFICIENT DOMINATION as the following property for a graph G and a vertex subset D:

$$\Pi := D \text{ is an efficient dominating set of } G.$$

The concrete gadgets are as follows: The initial-gadget $I(x)$ consists of a single edge, namely $i{-}x$. The copy-gadget $C(x,y)$ consists of a P_4, namely $x{-}s_1{-}s_2{-}y$. The one-in-three-gadget $O(x,y,z)$ consists of the union of three copies of the P_5, namely $a_1{-}a_2{-}\ldots{-}a_5$, $b_1{-}b_2{-}\ldots{-}b_5$, and $c_1{-}c_2{-}\ldots{-}c_5$, and three copies of the P_3, namely $a'_1{-}a'_2{-}a'_3$, $b'_1{-}b'_2{-}b'_3$, and $c'_1{-}c'_2{-}c'_3$, on $a_5 = b_5 = c_5$ (where $x = a_1$, $y = b_1$, and $z = c_1$) and three additional vertices a'', b'', and c'' that are adjacent exactly to a_2 and a'_2, b_2 and b'_2, and c_2 and c'_2 respectively. Figure 5.3 shows these gadgets.

Observation 87. *The initial-gadget fulfills the condition of Definition 18.*

Proof. Obviously, the initial-gadget allows exactly the efficient dominating sets $D = \{i\}$ and $D' = \{x\}$. □

Observation 88. *The copy-gadget fulfills the condition of Definition 19.*

Proof. Let $G = (V, E)$ be a graph with efficient dominating set D and let $x' \in V$. Furthermore, let $G' = (V', E')$ be the union of G and $C(x,y)$ on $x = x'$.

We show (1) and (2) simultaneously by constructing a Π-fulfilling set D' of G' from D with $x \in D \Leftrightarrow y \in D'$:

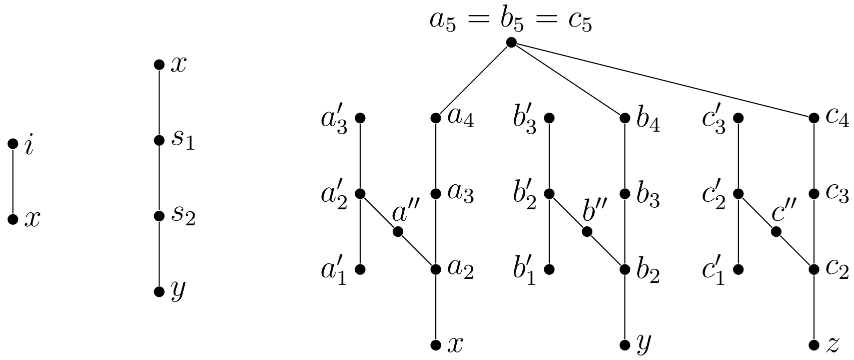

Figure 5.3: The initial-gadget $I(x)$, the copy-gadget $C(x, y)$, and the one-in-three-gadget $O(x, y, z)$ (from left to right) used for showing that EFFICIENT DOMINATION is NP-complete.

If $x' \in D$, then every efficient dominating set D' of G' with $D \subseteq D'$ contains x and dominates s_1, but it does not dominate s_2 with a vertex of D. Since D' must be efficient, this can only be done by $y \in D'$. Hence, $D' := D \cup \{y\}$ is the only efficient dominating set of G' with $D \subseteq D'$.

If $x' \notin D$, then every efficient dominating set D' of G' with $D \subseteq D'$ dominates x with a vertex of V, but it does not dominate s_1 with a vertex of D. This can only be done by $s_2 \in D'$, which means that D' dominates y with s_2. Hence, $D' := D \cup \{s_2\}$ is the only efficient dominating set of G' with $D \subseteq D'$. □

Observation 89. *The one-in-three-gadget fulfills the condition of Definition 21.*

Proof. Let $G = (V, E)$ be a graph and let $\{x', y', z'\} \subseteq V$. Furthermore, let $G' = (V', E')$ be the union of G and $O(x, y, z)$ on $x = x'$, $y = y'$, and $z = z'$.

(1): Let D' be an efficient dominating set of G'.

Notice that $\{a'_2, b'_2, c'_2\} \subseteq D'$ because there is no other way to dominate a'_1, a'_3, b'_1, b'_3, c'_1, and c'_3 without violating the efficiency of D'. This implies $\{a'', b'', c'', a_2, b_2, c_2\} \cap D' = \emptyset$.

Assume that more than one vertex of x, y, and z is in D'. Without loss of generality, say $\{x, y\} \subseteq D'$. This implies $\{a_2, a_3, b_2, b_3\} \cap D' = \emptyset$ by the efficiency of D'. Since D' must dominate a_3 and b_3, we have $\{a_4, b_4\} \subseteq D'$ because there are no other vertices that can dominate a_3 and b_3. But since both, a_4 and b_4, are adjacent to a_5, this contradicts the efficiency D'. Hence, at most one vertex of x, y, and z is in D'.

Assume that no vertex of x, y, and z is in D'. Since $\{a'', b'', c'', a_2, b_2, c_2\} \cap D' = \emptyset$, the only way to dominate a_2, b_2, and c_2 is $\{a_3, b_3, c_3\} \subseteq D'$. This makes it impossible to dominate a_5 without violating the efficiency of D'. Hence, D' contains exactly one vertex of x, y, and z.

(2): Let D be an efficient dominating set of G. If exactly one of the vertices x', y', and z' is in D, say $x' \in D$, then

$$D' := D \cup \{a_4, b_3, c_3, a'_2, b'_2, c'_2\}$$

clearly is an efficient dominating set of G'. □

Since the gadgets are trees and hence, bipartite, planar with only one face and cycle-free, the distance between x, y, and z is pairwise even in $O(x, y, z)$, the vertices x, y, and z have degree 1 in all gadgets, and the maximum degree inside the gadgets is at most 3, from Lemmas 82 to 85 and Corollary 86 follows:

Theorem 19. EFFICIENT DOMINATION *is NP-complete on planar bipartite graphs with maximum degree at most* 3 *and girth at least g, for every fixed g.*

5.3 Efficient Edge Domination

We can formulate EFFICIENT EDGE DOMINATION as the following property for a graph G and a vertex subset D:

$$\Pi := D \text{ is independent in } G \text{ and } G - D \text{ is 1-regular.}$$

The concrete gadgets are as follows: The initial-gadget $I(x)$ consists of a P_3, namely $i_1–i_2–x$. The copy-gadget $C(x, y)$ consists of a P_4, namely $x–s_1–s_2–y$. The one-in-three-gadget $O(x, y, z)$ consists of the union of three copies of the P_6, namely $a_1–a_2– \ldots –a_6$, $b_1–b_2– \ldots –b_6$, and $c_1–c_2– \ldots –c_6$, and three copies of the P_5, namely $a'_1–a'_2– \ldots –a'_5$, $b'_1–b'_2– \ldots –b'_5$, and $c'_1–c'_2– \ldots –c'_5$, on $a_6 = b_6 = c_6$ (where $x = a_1$, $y = b_1$, and $z = c_1$) and the additional edges $a_3a'_3$, $b_3b'_3$ and $c_3c'_3$. Figure 5.4 shows these gadgets.

Observation 90. *The initial-gadget fulfills the conditions of Definition 18.*

Proof. Clearly, there are exactly two different independent sets, D and D', such that $I(x) - D$ and $I(x) - D'$ respectively are 1-regular, namely $D = \{i_1\}$ and $D' = \{x\}$. □

Observation 91. *The copy-gadget fulfills the conditions of Definition 19.*

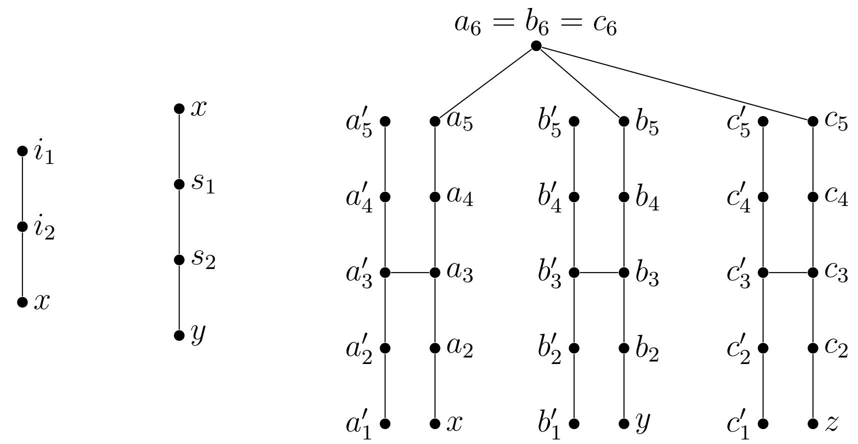

Figure 5.4: The initial-gadget $I(x)$, the copy-gadget $C(x, y)$ and the one-in-three-gadget $O(x, y, z)$ (from left to right) used for showing that EFFICIENT EDGE DOMINATION is NP-complete.

Proof. Let $G = (V, E)$ be a graph, let $x' \in V$, and let D be an independent vertex subset such that $G - D$ is 1-regular. Furthermore, let $G' = (V', E')$ be the union of G and $C(x, y)$ on $x = x'$.

We show (1) and (2) simultaneously by constructing a Π-fulfilling set D' of G' from D with $x \in D \Leftrightarrow y \in D'$:

Assume that $x' \in D$. Since the two neighbors s_1 and y of s_2 are not in D and $x{-}s_1$, every Π-fulfilling set D' of G' that fulfills $D \subseteq D'$ must contain either s_2 or y. If it contains s_2, then it cannot contain y and, thus, y is a vertex in $G' - D'$ of degree 0—this is a contradiction. Conversely, one can easily check that $D' := D \cup \{y\}$ is independent in G' and that $G' - D'$ is 1-regular because it equals $G - D$ with the additional isolated edge $s_1 s_2$. By definition, $y \in D'$.

Assume that $x' \notin D$. Since x' has degree 1 in $G - D$, every Π-fulfilling set D' of G' that fulfills $D \subseteq D'$ must contain s_1. This means that neither s_2 nor y can be in D' because otherwise either D' is not independent or $G' - D'$ is not 1-regular respectively. Clearly, $D' := D \cup \{s_1\}$ is independent in G' and $G' - D'$ is 1-regular because it equals $G - D$ with the additional isolated edge $s_2 y$. By definition, $y \notin D'$. □

Observation 92. *The one-in-three-gadget fulfills the conditions of Definition 21.*

Proof. Let $G = (V, E)$ be a graph and let $\{x', y', z'\} \subseteq V$. Furthermore, let $G' = (V', E')$ be the union of G and $O(x, y, z)$ on $x = x'$, $y = y'$, and $z = z'$.

(1): Let D' be a Π-fulfilling set of G'.

One can easily check that it must be $\{a'_3, b'_3, c'_3\} \subseteq D'$ and, hence, $\{a_3, b_3, c_3\} \cap D' = \emptyset$. Since $G' - D'$ is 1-regular, this implies that exactly one of a_2 and a_4, exactly one of b_2 and b_4, and exactly one of c_2 and c_4 is in D'.

Assume that at least two vertices of x, y, and z are in D'. Without loss of generality assume that $\{x, y\} \subseteq D'$. Since $x - a_2$ and $y - b_2$, by the foregoing considerations, we have $a_4 \in D'$ and $b_4 \in D'$, which implies $\{a_5, b_5\} \cap D' = \emptyset$. In both cases, $a_6 \in D'$ and $a_6 \notin D'$, $G' - D'$ is not 1-regular—this is a contradiction. Hence, at most one of x, y, and z is in D'.

Assume that $\{x, y, z\} \cap D' = \emptyset$. By the foregoing considerations, we have $\{a_2, b_2, c_2\} \subseteq D'$ because otherwise a_2, b_2, or c_2 has degree 2 in $G' - D'$. This implies $\{a_4, b_4, c_4\} \cap D' = \emptyset$. Since we already know that $\{a_3, b_3, c_3\} \cap D' = \emptyset$, we have $\{a_5, b_5, c_5\} \subseteq D'$ because otherwise a_4, b_4, or c_4 has degree 2 in $G' - D'$. This means that a_6 is an isolated vertex in $G' - D'$—this is a contradiction. Hence, at least one of x, y, z is in D'. This shows that exactly one of x, y, and z is in D'.

(2): Let D be a Π-fulfilling set of G. If exactly one vertex of x', y', z' is in D, say $x' \in D$, then

$$D' := D \cup \{a_4, a'_3, b_2, b'_3, b_5, c_2, c'_4, c_5\}$$

clearly is a Π-fulfilling set of G'. □

Since the gadgets are trees and hence, bipartite, planar with only one face and cycle-free, the distance between x, y, and z is pairwise even in $O(x, y, z)$, the vertices x, y, and z have degree 1 in all gadgets, and the maximum degree inside the gadgets is 3, from Lemmas 82 to 85 and Corollary 86 follows:

Theorem 20. EFFICIENT EDGE DOMINATION *is NP-complete on planar bipartite graphs with maximum degree at most* 3 *and girth at least* g*, for every fixed* g.

5.4 Polarity and Monopolarity

This section gives two NP-completeness results for MONOPOLARITY and POLARITY. In fact, the reduction framework does not work for POLARITY because polar graphs are not additive, but we later argue that the results hold for POLARITY as well.

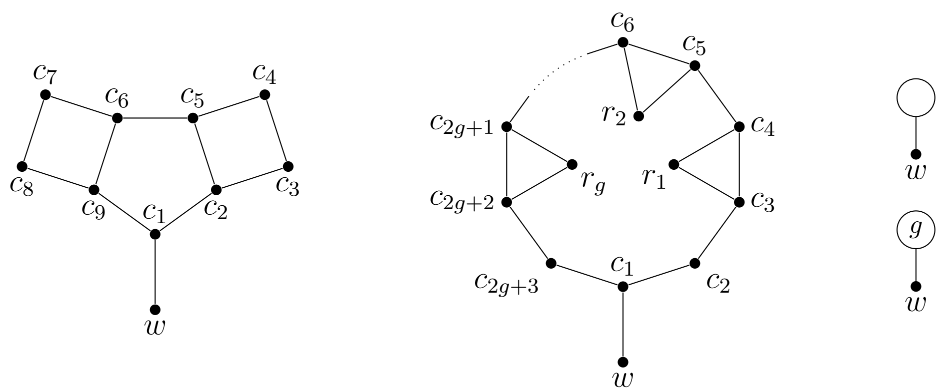

Figure 5.5: The helper gadgets H_1 (left) and H_2^g (center) and their symbolic drawings (right), where the upper pictograph represents H_1 and the lower pictograph represents H_2^g.

We can formulate MONOPOLARITY as the following property for a graph G and a vertex subset D:

$$\Pi := (D, V(G) \setminus D) \text{ is a monopolar partition of } G.$$

The concrete gadgets are as follows: The initial-gadget $I(x)$ simply consists of the vertex x. To define the copy- and the one-in-three-gadgets, we use a helper-gadget. We have to define two versions, H_1 and H_2^g, of the helper-gadget. The helper-gadget H_1 consists of the cycle $c_1{-}c_2{-}\ldots{-}c_9{-}c_1$ with the two chords c_2c_5 and c_6c_9 and a vertex w which is adjacent only to c_1. The helper-gadget H_2^g consists of the cycle $c_1{-}c_2{-}\ldots{-}c_{2g+3}$ and, for every $i \in \{1, \ldots, g\}$, the vertex r_i that is adjacent exactly to c_{2i+1} and c_{2i+2}, and a vertex w which is adjacent only to c_1. Figure 5.5 shows H_1 and H_2^g.

Observation 93. *In every monopolar partition of H_1 and H_2^g, the vertex w has a* `blue` *colored neighbor. Both, H_1 and H_2^g, admit a monopolar partition that colors w* `amber` *and a monopolar partition that colors w* `blue`*.*

Proof. It is easy to check that $H_1 - w$ admits exactly the monopolar partitions $(A, V(H_1 - w) \setminus A)$ with

$$\begin{aligned} A &= \{c_2, c_4, c_6, c_8\}\,, \\ A &= \{c_2, c_4, c_7, c_9\}\,, \text{ and} \\ A &= \{c_3, c_5, c_7, c_9\}\,. \end{aligned}$$

Similarly, one can easily verify that $H_2^g - w$ admits exactly the monopolar partitions $(A, V(H_2^g - w) \setminus A)$ with

$$A = \{c_2, c_4, \dots, c_{2g+2}\},$$
$$A = \{c_3, c_5, \dots, c_{2g+3}\},$$

and, for every $i \in \{1, \dots, k\}$,

$$A = \left\{c_2, c_4, \dots, c_{2(i-1)+2}, c_{2i+3}, c_{2(i+1)+3}, \dots, c_{2g+3}\right\}, \text{ and}$$
$$A = \left\{c_2, c_4, \dots, c_{2(i-1)+2}, r_i, c_{2i+3}, c_{2(i+1)+3}, \dots, c_{2g+3}\right\}.$$

In all these monopolar partitions, the vertex c_1 is `blue` colored. Hence, there are monopolar partitions of H_1, respectively H_2^g, that color w `amber` and w always has a `blue` colored neighbor. Notice that for both, $H_1 - w$ and $H_2^g - w$, there is a monopolar partition that colors all neighbors of c_1 `amber`. This monopolar partition clearly can be extended to a monopolar partition of H_1, respectively H_2^g, that colors w `blue`. □

The copy-gadget $C_1(x, y)$, respectively $C_2^g(x, y)$, is the union of the induced path $x - w' - y$ and H_1, respectively H_2^g, on $w = w'$. The one-in-three-gadget $O_1(x, y, z)$ is the union of five copies of the copy-gadget, $C_1(x, x')$, $C_1(y, y')$, $C_1(z, z')$, $C_1(x'', v)$, and $C_1(z'', w)$ on $x' = x''$ and $z' = z''$ and the additional edges $x'y'$, $y'z'$, and vw. The one-in-three-gadget $O_2^g(x, y, z)$ is the union of three copies of the copy-gadget, $C_2^g(x, x')$, $C_2^g(y, y')$, and $C_2^g(z, z')$ and two copies of the copy-chain-gadget, $CC_2^g(x'', v)$ and $CC_2^g(z'', w)$ on $x' = x''$ and $z' = z''$ and the additional edges $x'y'$, $y'z'$, and vw. Figure 5.6 shows these gadgets.

Observation 94. *The initial-gadget fulfills the condition of Definition 18.*

Proof. Obviously, the initial-gadget allows exactly the monopolar partitions $(\{x\}, \emptyset)$ and $(\emptyset, \{x\})$. □

Observation 95. *The copy-gadgets $C_1(x, y)$ and $C_2^g(x, y)$ fulfill the condition of Definition 19.*

Proof. Let $G = (V, E)$ be a graph with a monopolar partition $(D, V \setminus D)$ and let $x' \in V$. Furthermore, let $G' = (V', E')$ be the union of G and $C(x, y)$ on $x = x'$, where either $C(x, y) = C_1(x, y)$ or $C(x, y) = C_2^g(x, y)$. Let $(D', V' \setminus D')$ be a monopolar partition of G'.

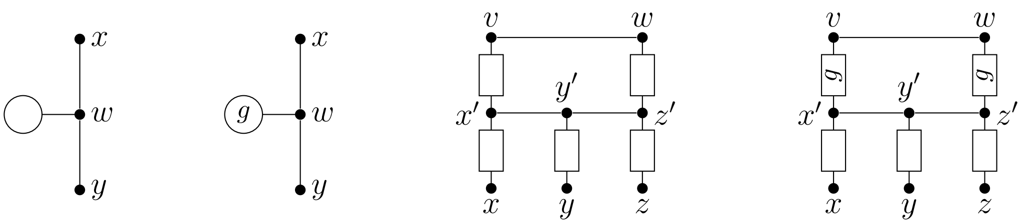

Figure 5.6: The copy-gadgets $C_1(x, y)$ and $C_2^g(x, y)$ and the one-in-three-gadgets $O_1(x, y, z)$ and $O_2^g(x, y, z)$ (from left to right) to show that MONOPOLARITY is NP-complete. Notice that the one-in-three gadgets contain copies of the copy-gadgets and copy-chain-gadgets using the pictographs from Figure 5.1. Clearly, the pictographs in $O_1(x, y, z)$ represent $C_1(x, y)$ and the pictographs in $O_2^g(x, y, z)$ represent $C_2^g(x, y)$ and $CC_2^g(x, y)$ respectively.

(1): If $x \in D'$, then clearly $w' \notin D'$. Since, by Observation 93, w' has a `blue` neighbor, it must be $y \in D'$ because otherwise y, w', and the `blue` neighbor of w' form an induced P_3 in $G'[V' \setminus D']$.

If $x \notin D'$, then we have $w' \in D'$ because otherwise x, w', and the `blue` colored neighbor of w' in the helper-gadget forms an induced P_3 in $G'[V' \setminus D']$. Thus, since D' is independent, we have $y \notin D'$.

(2): By Observation 93, w' can be colored `amber` and w' can be colored `blue` in the helper gadget. Since monopolar graphs are hereditary, this implies that every monopolar partition of G can be extended to a monopolar partition of G'. □

Observation 96. *The one-in-three-gadgets $O_1(x, y, z)$ and $O_2^g(x, y, z)$ fulfill the condition of Definition 21.*

Proof. For both, $O_1(x, y, z)$ and $O_2^k(x, y, z)$, Observation 95 implies that every monopolar partition colors x, x', v with the same color and z, z', w with the same color and y, y' with the same color. Since v and w are adjacent, no monopolar partition colors both, v and w `amber` and, hence, no monopolar partition colors x' and z' `amber`. Since x'–y'–z' is an induced P_3, at least one of its vertices is colored `amber`. The only way to color more than one vertex `amber`, namely coloring x' and z' `amber` and y' `blue`, is already shown to be impossible. Hence, every monopolar partition colors exactly one vertex of x', y', and z' `amber`, that is, exactly one vertex of x, y, and z. It is easily checked that there is a monopolar partition for all of the three choices.

Let $G = (V, E)$ be a graph with $\{x', y', z'\} \subseteq V$ and let $(D, V \setminus D)$ be a monopolar partition of G. Furthermore, let $G' = (V', E')$ be the union of G and $O(x,y,z)$ on $x = x'$, $y = y'$, and $z = z'$, where either $O(x,y,z) = O_1(x,y,z)$ or $O(x,y,z) = O_2^k(x,y,z)$.

(1): Let $(D', V' \setminus D')$ be a monopolar partition of G'. By the foregoing considerations, every monopolar partition of G' colors exactly one vertex of x', y', and z' `blue`, that is, exactly one of x', y', and z' is in D'.

(2): Since monopolar graphs are hereditary, it is easy to verify that, for every monopolar partition $(D, V \setminus D)$ of G with $|\{x', y', z'\} \cap D| = 1$, there is a monopolar partition $(D', V' \setminus D')$ of G' with $D \subseteq D'$. $\square$

Since the gadgets are planar with all connection vertices on the outer face, the vertices x, y, and z have degree 1 in all gadgets, the maximum degree inside the gadgets is at most 3, $C_1(x,y)$ and $O_1(x,y,z)$ are triangle-free, and $C_2^g(x,y)$ and $O_2^g(x,y,z)$ are $\{C_4, \ldots, C_g\}$-free, from Lemmas 82, 84, and 85 and Corollary 86 follows:

Theorem 21. MONOPOLARITY *is NP-complete on planar triangle-free graphs with maximum degree at most* 3 *and on planar* $\{C_4, \ldots, C_g\}$*-free graphs with maximum degree at most* 3*, for every fixed* $g \geq 4$.

One can easily check that all gadgets admit only polar partitions that are also monopolar partitions. This is also true for the reduction graph $G(F)$ described in Section 5.1.3. As a simple argument for that, consider a polar partition (A, B) of $G(F)$ that is not monopolar, that is, $G[A]$ is a complete multipartite graph with at least two independent sets. The degree restriction on $G(F)$, that is, every vertex has degree at most 3, implies that $|A| \leq 6$. But it is quite obvious that this is not sufficient to meet every induced P_3 of $G(F)$. Hence, Theorem 21 also holds for POLARITY. Furthermore, the class of polar graphs is self-complementary, that is, a graph is polar, if and only if its complement is polar. Hence, the results can be extended to the complements of the mentioned graph classes:

Theorem 22. POLARITY *is NP-complete on*

- *planar triangle-free graphs with maximum degree at most* 3*,*
- *co-planar* $3P_1$*-free* n*-vertex graphs with minimum degree at least* $n - 4$*,*
- *planar* $\{C_4, \ldots, C_g\}$*-free graphs with maximum degree at most* 3*, and*
- *co-planar* $\left\{\overline{C_4}, \ldots, \overline{C_g}\right\}$*-free* n*-vertex graphs with minimum degree at least* $n - 4$*,*

for every fixed $g \geq 4$.

Since every K_4-free graph with maximum degree at most 3 is 3-colorable by Brooks' Theorem, Theorems 21 and 22 imply:

Corollary 97. MONOPOLARITY *and* POLARITY *are NP-complete on* 3-*colorable graphs.* POLARITY *is NP-complete on claw-free co-planar graphs and on* $\{2K_2, C_5\}$*-free graphs and, therefore, on hole-free graphs and* P_5*-free graphs.*

6 Conclusion

This thesis advances the computational complexity analysis for EFFICIENT DOMINATION, EFFICIENT EDGE DOMINATION, POLARITY, and MONOPOLARITY on restricted graph classes.

For EFFICIENT DOMINATION on F-free graphs, that is, the graph classes that are characterizable by a single forbidden induced subgraph F, we strive after a perfect dichotomy between the tractable cases and those where the problem is NP-complete. We point out that for all tractable cases, F is a linear forest. Chapter 2 succeeds in classifying the graph classes where F has at most six vertices, except for the P_6-free graphs. Since the problem is NP-complete on $(P_3 + P_3)$-free graphs, the remaining open cases are:

- P_6-free graphs,
- $(P_5 + P_2)$-free graphs,
- $(P_6 + P_2)$-free graphs,
- $(P_k + mP_2)$-free graphs, for every $3 \leq k \leq 6$ and $m \geq 2$.

For the case of P_6-free graphs, some progress has been made by Friese [46]. He shows that the squares of efficiently dominatable P_6-free graphs are hole-free and conjectures that they are even perfect. If this conjecture is true, the reduction technique presented in Section 2.3 will immediately imply a polynomial-time algorithm because MAXIMUM WEIGHT INDEPENDENT SET is polynomially-time solvable on perfect graphs.

In this thesis, we present an $O(nm)$-time algorithm for EFFICIENT EDGE DOMINATION on hole-free graphs. We leave it for future work whether EFFICIENT EDGE DOMINATION remains tractable when the C_5 or even longer cycles are allowed. Since the problem is linear-time solvable on chordal bipartite graphs, as shown in Section 3.1, on chordal graphs [70], and P_7-free graphs [14], one may ask if there is also a linear-time algorithm for hole-free graphs.

As far as we know, our tractability result for MONOPOLARITY, namely the efficient algorithm for MONOPOLAR EXTENSION on locally A_5-$S_{2,2,2}$-defused graphs, covers all known tractable cases except some graph classes of bounded tree- or clique-width. In our opinion, the reduction to 2-SATISFIABILITY can hardly be improved to work for significantly larger or different graph classes. Hence, to find new tractable cases for MONOPOLARITY, we think that it is necessary to develop new techniques.

Since many of the known efficient algorithms for POLARITY solve MONOPOLARITY or even MONOPOLAR EXTENSION as a subroutine, finding new approaches for MONOPOLARITY is of particular interest with regard to POLARITY. In particular, by the algorithmic framework of Section 4.4, every tractable case for MONOPOLAR EXTENSION on a hereditary graph class $\mathcal{C}$ implies an efficient algorithm for POLARITY on planar $\mathcal{C}$ graphs. This may lead to the opinion that the complexities of MONOPOLARITY and POLARITY are highly related, at least on planar graphs. In fact, we are not aware of any graph class where POLARITY is tractable and MONOPOLARITY is NP-complete. Conversely, there are graph classes, for example hole-free graphs, on which POLARITY is NP-complete, as we show in Section 5.4, but MONOPOLARITY is efficiently solvable, as we show in Section 4.3.

We hope that this thesis gives new insights into the studied problems and helps to initiate future work on their classification.

Bibliography

[1] S. Arnborg, J. Lagergren, and D. Seese. “Easy problems for tree-decomposable graphs”. In: *Journal of Algorithms* 12.2 (1991), pp. 308–340.

[2] B. Aspvall, M. F. Plass, and R. E. Tarjan. “A linear-time algorithm for testing the truth of certain quantified boolean formulas”. In: *Information Processing Letters* 8.3 (1979). (Erratum in IPL 14.4 (1982)), pp. 121–123.

[3] E. Balas and C. S. Yu. “On graphs with polynomially solvable maximum-weight clique problem”. In: *Networks* 19.2 (1989), pp. 247–253.

[4] D. W. Bange, A. E. Barkauskas, and P. J. Slater. “Efficient dominating sets in graphs”. In: *Applications of Discrete Mathematics* (1988). Proceedings in Applied Mathematics Series (33), pp. 189–199.

[5] L. W. Beineke. *On derived graphs and digraphs.* Beitr. Graphentheorie, Int. Kolloquium Manebach (DDR). 1968.

[6] N. Biggs. “Perfect codes in graphs”. In: *Journal of Combinatorial Theory, Series B* 15.3 (1973), pp. 289–296.

[7] A. Brandstädt, P. Fičur, A. Leitert, and M. Milanič. “Polynomial-time Algorithms for Weighted Efficient Domination Problems in AT-free Graphs and Dually Chordal Graphs”. Manuscript. 2013.

[8] A. Brandstädt, C. Hundt, and R. Nevries. “Efficient Edge Domination on Hole-Free Graphs in Polynomial Time”. In: *LATIN 2010: Theoretical Informatics*. Ed. by A. López-Ortiz. Vol. 6034. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 650–661.

[9] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999.

[10] A. Brandstädt, A. Leitert, and D. Rautenbach. “Efficient Dominating and Edge Dominating Sets for Graphs and Hypergraphs”. In: *Algorithms and Computation*. Ed. by K.-M. Chao, T.-s. Hsu, and D.-T. Lee. Vol. 7676. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 267–277.

[11] A. Brandstädt, A. Leitert, and D. Rautenbach. “Efficient Dominating and Edge Dominating Sets for Graphs and Hypergraphs”. In: *CoRR* abs/1207.0953 (2012).

[12] A. Brandstädt, M. Milanič, and R. Nevries. “New Polynomial Cases of the Weighted Efficient Domination Problem”. In: *MFCS*. Ed. by K. Chatterjee and J. Sgall. Vol. 8087. Lecture Notes in Computer Science. Springer, 2013, pp. 195–206.

[13] A. Brandstädt, M. Milanič, and R. Nevries. “New Polynomial Cases of the Weighted Efficient Domination Problem”. In: *CoRR* abs/1304.6255 (2013).

[14] A. Brandstädt and R. Mosca. “Dominating Induced Matchings for P_7-Free Graphs in Linear Time”. In: *Algorithmica* (2012), pp. 1–21.

[15] J. I. Brown and D. G. Corneil. “On generalized graph colorings”. In: *Journal of Graph Theory* 11.1 (1987), pp. 87–99.

[16] D. M. Cardoso, J. O. Cerdeira, C. Delorme, and P. C. Silva. “Efficient edge domination in regular graphs”. In: *Discrete Applied Mathematics* 156.15 (2008), pp. 3060–3065.

[17] D. M. Cardoso, N. Korpelainen, and V. V. Lozin. “On the complexity of the dominating induced matching problem in hereditary classes of graphs”. In: *Discrete Applied Mathematics* 159.7 (2011), pp. 521–531.

[18] D. M. Cardoso and V. V. Lozin. “Dominating Induced Matchings”. In: *Graph Theory, Computational Intelligence and Thought*. Ed. by M. Lipshteyn, V. Levit, and R. McConnell. Vol. 5420. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 77–86.

[19] G. Chang, C. Rangan, and S. Coorg. “Weighted independent perfect domination on cocomparability graphs”. In: *Discrete Applied Mathematics* 63.3 (1995), pp. 215–222.

[20] M.-S. Chang and Y.-C. Liu. “Polynomial Algorithms for the Weighted Perfect Domination Problems on Chordal Graphs and Split Graphs”. In: *Inf. Process. Lett.* 48.4 (1993), pp. 205–210.

[21] M.-S. Chang and Y.-C. Liu. "Polynomial Algorithms for Weighted Perfect Domination Problems on Interval and Circular-Arc Graphs." In: *J. Inf. Sci. Eng.* 11.4 (1994), pp. 549–568.

[22] Z. A. Chernyak and A. A. Chernyak. "About recognizing (α, β) classes of polar". In: *Discrete Mathematics* 62.2 (1986), pp. 133–138.

[23] R. Churchley. "Some Remarks on Monopolarity". Manuscript. 2012.

[24] R. Churchley and J. Huang. "Line-Polar Graphs: Characterization and Recognition". In: *SIAM Journal on Discrete Mathematics* 25.3 (2011), pp. 1269–1284.

[25] R. Churchley and J. Huang. "List monopolar partitions of claw-free graphs". In: *Discrete Mathematics* 312.17 (2012), pp. 2545–2549.

[26] R. Churchley and J. Huang. "On the Polarity and Monopolarity of Graphs". In: *Journal of Graph Theory* (2013). eprint: `http://dx.doi.org/10.1002/jgt.21755`.

[27] R. Churchley and J. Huang. "Solving Partition Problems with Colour-Bipartitions". In: *Graphs and Combinatorics* (2012). eprint: `http://dx.doi.org/10.1007/s00373-012-1270-z`.

[28] S. A. Cook. "The Complexity of Theorem-Proving Procedures". In: *STOC*. Ed. by M. A. Harrison, R. B. Banerji, and J. D. Ullman. ACM, 1971, pp. 151–158.

[29] D. Coppersmith and S. Winograd. "Matrix Multiplication via Arithmetic Progressions". In: *J. Symb. Comput.* 9.3 (1990), pp. 251–280.

[30] D. G. Corneil, Y. Perl, and L. K. Stewart. "A Linear Recognition Algorithm for Cographs". In: *SIAM J. Comput.* 14.4 (1985), pp. 926–934.

[31] B. Courcelle. "The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues". In: *ITA* 26 (1992), pp. 257–286.

[32] B. Courcelle, J. A. Makowsky, and U. Rotics. "Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width". In: *Theory Comput. Syst.* 33.2 (2000), pp. 125–150.

[33] B. Courcelle and M. Mosbah. "Monadic Second-Order Evaluations on Tree-Decomposable Graphs". In: *Theor. Comput. Sci.* 109.1-2 (1993), pp. 49–82.

[34] M. Davis and H. Putnam. "A Computing Procedure for Quantification Theory". In: *J. ACM* 7.3 (1960), pp. 201–215.

[35] T. Ekim, P. Heggernes, and D. Meister. "Polar permutation graphs are polynomial-time recognisable". In: *European Journal of Combinatorics* 34.3 (2013), pp. 576–592.

[36] T. Ekim, P. Hell, J. Stacho, and D. de Werra. "Polarity of chordal graphs". In: *Discrete Applied Mathematics* 156.13 (2008), pp. 2469–2479.

[37] T. Ekim and J. Huang. "Recognizing line-polar bipartite graphs in time". In: *Discrete Applied Mathematics* 158.15 (2010), pp. 1593–1598.

[38] T. Ekim, N. V. R. Mahadev, and D. de Werra. "Polar cographs". In: *Discrete Applied Mathematics* 156.10 (2008), pp. 1652–1660.

[39] E. M. Eschen and X. Wang. "Algorithms for unipolar and generalized split graphs". In: *Discrete Applied Mathematics* 162 (2014), pp. 195–201.

[40] S. Even, A. Itai, and A. Shamir. "On the Complexity of Timetable and Multicommodity Flow Problems". In: *SIAM Journal on Computing* 5.4 (1976), pp. 691–703.

[41] M. Farber. "On diameters and radii of bridged graphs". In: *Discrete Mathematics* 73.3 (1989), pp. 249–260.

[42] A. Farrugia. "Vertex-Partitioning into Fixed Additive Induced-Hereditary Properties Is NP-hard". In: *Electronic J. of Combinatorics* 11 (2004).

[43] T. Feder, P. Hell, S. Klein, and R. Motwani. "Complexity of Graph Partition Problems". In: *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*. STOC '99. New York, NY, USA: ACM, 1999, pp. 464–472.

[44] T. Feder, P. Hell, S. Klein, and R. Motwani. "List Partitions". In: *SIAM Journal on Discrete Mathematics* 16.3 (2003), pp. 449–478.

[45] M. Fellows and M. Hoover. "Perfect Domination". In: *Australasian J. of Combinatorics* 3 (1991), pp. 141–150.

[46] E. Friese. "Das Efficient-Domination-Problem auf P_6-freien Graphen". German. MA thesis. University of Rostock, 2013.

[47] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[48] M. R. Garey and D. S. Johnson. “The Rectilinear Steiner Tree Problem is NP-Complete”. In: *SIAM Journal on Applied Mathematics* 32.4 (1977), pp. 826–834.

[49] D. L. Grinstead, P. J. Slater, N. A. Sherwani, and N. D. Holmes. “Efficient Edge Domination Problems in Graphs”. In: *Inf. Process. Lett.* 48.5 (1993), pp. 221–228.

[50] M. Grötschel, L. Lovász, and A. Schrijver. “Polynomial Algorithms for Perfect Graphs”. In: *Topics on Perfect Graphs*. Ed. by C. Berge and V. Chvátal. Vol. 88. North-Holland Mathematics Studies. North-Holland, 1984, pp. 325–356.

[51] T. Haynes, S. Hedetniemi, and P. Slater. *Domination in Graphs: Advanced Topics*. Chapman and Hall/CRC Pure and Applied Mathematics Series. Marcel Dekker, Incorporated, 1998.

[52] T. Haynes, S. Hedetniemi, and P. Slater. *Fundamentals of Domination in Graphs*. Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis, 1998.

[53] R. B. Hayward. “Two classes of perfect graphs”. PhD thesis. McGill University, Montreal, Canada, 1988.

[54] P. Heggernes and J. A. Telle. “Partitioning Graphs into Generalized Dominating Sets”. In: *Nord. J. Comput.* 5.2 (1998), pp. 128–142.

[55] M. A. Henning. “A survey of selected recent results on total domination in graphs”. In: *Discrete Mathematics* 309.1 (2009), pp. 32–63.

[56] J. E. Hopcroft and R. E. Tarjan. “Efficient Algorithms for Graph Manipulation”. In: *Commun. ACM* 16.6 (1973), pp. 372–378.

[57] J. Huang and B. Xu. “A forbidden subgraph characterization of line-polar bipartite graphs”. In: *Discrete Applied Mathematics* 158.6 (2010), pp. 666–680.

[58] R. M. Karp. “Reducibility Among Combinatorial Problems”. In: *Complexity of Computer Computations*. Ed. by R. E. Miller and J. W. Thatcher. The IBM Research Symposia Series. Plenum Press, New York, 1972, pp. 85–103.

[59] N. Korpelainen. “A Polynomial-time Algorithm for the Dominating Induced Matching Problem in the Class of Convex Graphs”. In: *Electronic Notes in Discrete Mathematics* 32.0 (2009), pp. 133–140.

[60] N. Korpelainen, V. V. Lozin, and C. Purcell. “Dominating induced matchings in graphs without a skew star”. In: *Journal of Discrete Algorithms* (2013). eprint: `http://dx.doi.org/10.1016/j.jda.2013.11.002`.

[61] P. Laroche. “Planar 1-in-3 satisfiability is NP-complete”. In: *Comtes rendus de l’academie des sciences serie I-mathematique* 316.4 (1993), pp. 389–392.

[62] H.-O. Le and V. B. Le. “The NP-completeness of (1,r)-subcolorability of cubic graphs”. In: *Information Processing Letters* 81.3 (2002), pp. 157–162.

[63] V. B. Le and R. Nevries. “Complexity and algorithms for recognizing polar and monopolar graphs”. In: *Theor. Comput. Sci.* (2014). eprint: `http://dx.doi.org/10.1016/j.tcs.2014.01.032`.

[64] V. B. Le and R. Nevries. “Recognizing Polar Planar Graphs Using New Results for Monopolarity”. In: *Algorithms and Computation*. Ed. by T. Asano, S.-i. Nakano, Y. Okamoto, and O. Watanabe. Vol. 7074. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 120–129.

[65] Y. D. Liang, C. L. Lu, and C. Y. Tang. “Efficient domination on permutation graphs and trapezoid graphs”. In: *Computing and Combinatorics*. Ed. by T. Jiang and D. Lee. Vol. 1276. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1997, pp. 232–241.

[66] R. Lin and S. Olariu. “An NC recognition algorithm for cographs”. In: *Journal of Parallel and Distributed Computing* 13.1 (1991), pp. 76–90.

[67] Y.-L. Lin. “Fast Algorithms for Independent Domination and Efficient Domination in Trapezoid Graphs”. In: *ISAAC*. 1998, pp. 267–276.

[68] M. Livingston and Q. F. Stout. “Distributing Resources in Hypercube Computers”. In: *Proceedings of the Third Conference on Hypercube Concurrent Computers and Applications: Architecture, Software, Computer Systems, and General Issues - Volume 1*. C3P. New York, NY, USA: ACM, 1988, pp. 222–231.

[69] V. V. Lozin and M. Milanič. “A polynomial algorithm to find an independent set of maximum weight in a fork-free graph”. In: *J. Discrete Algorithms* 6.4 (2008), pp. 595–604.

[70] C. L. Lu, M.-T. Ko, and C. Y. Tang. “Perfect edge domination and efficient edge domination in graphs”. In: *Discrete Applied Mathematics* 119.3 (2002), pp. 227–250.

[71] C. L. Lu and C. Y. Tang. “Efficient domination in bipartite graphs”. Manuscript. 1997.

[72] C. L. Lu and C. Y. Tang. “Solving the Weighted Efficient Edge Domination Problem on Bipartite Permutation Graphs”. In: *Discrete Applied Mathematics* 87.1-3 (1998), pp. 203–211.

[73] C. L. Lu and C. Y. Tang. “Weighted efficient domination problem on some perfect graphs”. In: *Discrete Applied Mathematics* 117.1-3 (2002), pp. 163–182.

[74] A. Lubiw. “Doubly Lexical Orderings of Matrices”. In: *SIAM J. Comput.* 16.5 (1987), pp. 854–879.

[75] R. M. McConnell and J. Spinrad. “Modular decomposition and transitive orientation”. In: *Discrete Mathematics* 201.1-3 (1999), pp. 189–241.

[76] M. Milanič. “Hereditary Efficiently Dominatable Graphs”. In: *Journal of Graph Theory* (2012). eprint: `http://dx.doi.org/10.1002/jgt.21685`.

[77] G. J. Minty. “On maximal independent sets of vertices in claw-free graphs”. In: *Journal of Combinatorial Theory, Series B* 28.3 (1980), pp. 284–304.

[78] C. Moore and J. M. Robson. “Hard Tiling Problems with Simple Tiles”. In: *Discrete & Computational Geometry* 26.4 (2001), pp. 573–590.

[79] W. Mulzer and G. Rote. “Minimum-weight triangulation is NP-hard”. In: *J. ACM* 55.2 (2008), Article 11.

[80] D. Nakamura and A. Tamura. “A revision of Minty’s algorithm for finding a maximum weight stable set of a claw-free graph”. In: *J. Oper. Res. Soc. Japan* 44.2 (2001), pp. 194–204.

[81] S. D. Nikolopoulos and L. Palios. “Detecting Holes and Antiholes in Graphs”. In: *Algorithmica* 47.2 (2007), pp. 119–138.

[82] G. Oriolo, U. Pietropaoli, and G. Stauffer. “A New Algorithm for the Maximum Weighted Stable Set Problem in Claw-Free Graphs”. In: *Integer Programming and Combinatorial Optimization*. Ed. by A. Lodi, A. Panconesi, and G. Rinaldi. Vol. 5035. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 77–96.

[83] R. Paige and R. E. Tarjan. “Three Partition Refinement Algorithms”. In: *SIAM J. Comput.* 16.6 (1987), pp. 973–989.

[84] S. Poljak. “A note on stable sets and colorings of graphs.” In: *Commentat. Math. Univ. Carol.* 15 (1974), pp. 307–309.

[85] E. Prisner. “Graphs with few cliques.” In: *Graph theory, combinatorics, algorithms and applications. Vol. 2. Proceedings of the seventh quadrennial international conference on the theory and applications of graphs, Kalamazoo, USA,* New York: Wiley, 1995, pp. 945–956.

[86] V. Raghavan and J. Spinrad. “Robust algorithms for restricted domains”. In: *J. Algorithms* 48.1 (2003), pp. 160–172.

[87] J. M. Schmidt. “A simple test on 2-vertex- and 2-edge-connectivity”. In: *Inf. Process. Lett.* 113.7 (2013), pp. 241–244.

[88] J. Spinrad. “Doubly Lexical Ordering of Dense 0-1 Matrices”. In: *Inf. Process. Lett.* 45.5 (1993), pp. 229–235.

[89] J. Spinrad. “Finding Large Holes”. In: *Inf. Process. Lett.* 39.4 (1991), pp. 227–229.

[90] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. “A New Algorithm for Generating All the Maximal Independent Sets”. In: *SIAM J. Comput.* 6.3 (1977), pp. 505–517.

[91] R. I. Tyshkevich and A. A. Chernyak. “Algorithms for the canonical decomposition of a graph and recognizing polarity”. Russian. In: *Izvestia Akad. Nauk BSSR, Ser. Fiz. Mat. Navuk.* 6 (1985), pp. 16–23.

[92] R. I. Tyshkevich and A. A. Chernyak. “Decomposition of graphs”. In: *Cybernetics* 21.2 (1985), pp. 231–242.

[93] V. V. Williams. “Multiplying matrices faster than Coppersmith-Winograd”. In: *STOC.* Ed. by H. J. Karloff and T. Pitassi. ACM, 2012, pp. 887–898.

[94] C.-C. Yen and R. C. T. Lee. “The Weighted Perfect Domination Problem and Its Variants”. In: *Discrete Applied Mathematics* 66.2 (1996), pp. 147–160.