Domination Problems on Special Graph Classes

Dissertation

zur
Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Ingenieurwissenschaften

der Universität Rostock

vorgelegt von

Thomas Szymczak, geb. am 13.03.1971 in Dinslaken (NRW)

aus Rostock

Rostock, 07.12.2001

Datum der Verteidigung: 18. Oktober 2002

Gutachter

Prof. Dr. Andreas Brandstädt
Universität Rostock
Fachbereich Informatik
18051 Rostock
Deutschland

Prof. Dr. Dieter Kratsch
Université de Metz
UFR MIM
Département d'informatique
Ile du Saulcy
57045 Metz Cedex 1
France

Prof. Dr. Feodor F. Dragan
Kent State University
Department of Computer Science
Kent, Ohio 44242
U.S.A.

Vorwort

Die höchste Klugheit besteht darin,
den Wert der Dinge genau zu kennen.
(François Duc de La Rochefoucauld)

Viele Probleme, die sich mit dem strategischen Plazieren von irgendwelchen Objekten in einem Netzwerk beschäftigen, lassen sich als Dominationsprobleme in gewissen Graphen beschreiben. Die Wurzeln von Domination gehen bis in das Jahr 1850 zurück, wo sich NAUCK, GAUSS ET AL. mit dem Plazieren von Schachfiguren auf einem $n \times n$–Brett beschäftigten. Ziel war es, möglichst wenig Figuren so auf dem Brett zu plazieren, daß alle Felder dominiert werden.

1975 wurde dann mit der algorithmischen Untersuchung des Problems MINIMUM DOMINATING SET begonnen. JOHNSON war der erste, der die $\mathbb{NP}$–Vollständigkeit dieses Problems für allgemeine Graphen erkannte; COCKAYNE ET AL. entwickelten den ersten Linearzeitalgorithmus für die Klasse der Bäume. Seitdem sind unzählige Arbeiten über Domination entstanden.

1995 habe ich bei der Erstellung meiner Diplomarbeit den Zugang zu Domination gefunden. Mein Interesse galt insbesondere der Entwicklung von effizienten Algorithmen für möglichst große Graphenklassen. Diese Arbeit setzte ich während meiner Tätigkeit als Wissenschaftlicher Assistent am Institut für Theoretische Informatik an der Universiät Rostock fort, wo auch diese Abhandlung entstand.

Die Resultate stammen vorwiegend aus folgenden Arbeiten:

- A. BRANDSTÄDT, T. KLEMBT, V.B. LE, S. MAHFUD, T. SZYMCZAK, On the structure and clique width of graph classes defined by two forbidden four-vertex graphs, *Manuskript*, 2001.
- F. NICOLAI, T. SZYMCZAK, r–Domination problems on trees and their homogeneous extensions, *International Journal of Mathematical Algorithms* 1 (1999), 53–79.
- F. NICOLAI, T. SZYMCZAK, Homogeneous sets and domination: A linear time algorithm for distance–hereditary graphs, *Networks* 37(3) (2001), 117–128.

Neben meinen Coautoren danke ich besonders PD Dr. Van Bang Le und Prof. Dr. Andreas Brandstädt für die wertvollen Korrekturhinweise und die vielen fruchtbaren Diskussionen.

Rostock, im Dezember 2001 *Thomas Szymczak*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Location problems play an important role in network design. Let G be a network structure and let processors of two types (suppliers and receivers, for modelling a coverage problem in telecommunication) be assigned to the vertices of G. To each receiver a value is associated indicating the radius within which it can receive information from suppliers. Assuming that the production effort for suppliers is much higher than for receivers we have to minimize the number of suppliers which are necessary to provide all receivers with information. If all receivers are of the same type, i.e. their radius value is identical, then this is exactly the well–known k–domination problem on graphs (cf. [42]): Compute a minimum cardinality set D such that for each vertex v outside D there is at least one vertex inside D of distance at most k to v. Assigning a supplier to each vertex of D then minimizes their number. If we allow different types of receivers, i.e. the radii do not coincide, then we have the more general r–domination problem: Given a graph G and a radius function $r : V(G) \to \mathbb{N}$ compute a minimum cardinality set D such that for each vertex v outside D there is at least one vertex inside D of distance at most $r(v)$. Note that $r(v) = 0$ means that this vertex must belong to D. So we can extend a given substructure of the network by assigning radius zero to already installed suppliers.

It is well–known that the domination problem is $\mathbb{NP}$–complete for general graphs (JOHNSON, circa 1975). In recent years the behaviour of certain graph classes with respect to (several) domination problems were investigated. Hereby, the following two approaches play an important role for solving domination problems on a graph class $\mathfrak{G}$:

- The design of a tree representation for $\mathfrak{G}$. Very often efficient algorithms for domination can be developed by using an underlying tree structure of a graph (cf. [9, 47, 55]). In [47] the authors review the complexity of the minimum dominating set problem on several families of perfect graphs using their tree representations.

- Shrinking homogeneous sets to smaller components and then, recursively solve the problem (cf. [54, 108]).

A combination of both approaches was used in [108] for solving the r–dominating set problem on homogeneous extensions of trees in linear time. Hereby, a graph is a homogeneous extension of a tree iff the reduction of all homogeneous sets to single vertices gives a tree.

More general, we obtained in [108], that efficient algorithms solving the r–dominating clique and the connected r–dominating set problem on a hereditary graph class $\mathfrak{G}$ lead to efficient algorithms on their homogeneous extensions, too.

In this thesis we extend the results of [108] regarding the r–dominating set problem to a superclass of homogeneous extensions, namely $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$.

Recently, the concept of clique width attracted much attention. In [40] the authors give an unified approach to get efficient solutions for many algorithmic graph problems on graph classes of bounded clique width. One condition needed for this is that the given problem is expressible in terms of a logical expression in a so–called Monadic Second Order Logic. MINIMUM DOMINATING SET and most of the variants of this problem are such problems therefore bounded clique width is an important method solving domination problems.

Many hereditary graph classes can be characterized by forbidden induced subgraphs or, if not, at least a list of small forbidden subgraphs is known (see [23] for an overview of such classes). Thus, it is interesting to consider $\mathcal{F}$–free graphs where $\mathcal{F}$ is a set of graphs containing at most four vertices. For all of these classes we investigate if MINIMUM DOMINATING SET is $\mathbb{NP}$–complete or if it can be solved by a polynomial time algorithm.

The paper is organized as follows.

At first we give basic definitions and notions (chapter 2).

After that in chapter 3 we work on domination problems on special graph classes. Initially, we consider graph classes of bounded clique width. We investigate which domination problems can be expressed in Monadic Second Order Logic (see Theorem 3.1.3) and prove that the clique width of (K_4,co-paw)-free graphs (resp. (K_4,diamond,C_4,claw)–free graphs) is bounded (resp. unbounded). In [20] we extend this to nearly all combinations of graph classes defined by exactly two forbidden four–vertex graphs.

Using these results we consider $\mathcal{F}$–free graphs where $\mathcal{F}$ is a set of graphs containing at most four vertices. For all of these classes we investigate if MINIMUM DOMINATING SET is $\mathbb{NP}$–complete or if it can be solved by a polynomial time algorithm (see Corollary 3.2.14).

Next, we consider the relation of homogeneous sets to the minimum r–dominating set problem. We show that one can reduce a homogeneous set to one vertex or to a so–called meta–vertex consisting of two nonadjacent vertices with r–value one. This reduction together with modular decomposition leads to an $O(|V||E|)$ time algorithm for computing a minimum r–dominating set for graphs which can be generated from the one–vertex graph by a finite number of homogeneous extensions (substitution of an arbitrary graph into a vertex) and by attaching pendant vertices (leaves). For distance–hereditary graphs — a proper subclass of this graph class — we even get a linear time algorithm.

In the appendix we list the complexity of some domination problems on different graph classes (but we declare no claim to completeness). This information can be a starting point to extend the information system on graph class inclusions (ISGCI, `http://www.informatik.uni-rostock.de/~gdb/isgci/Isgci.html`) with information about the complexity of important graph theoretic problems.

Chapter 2

Preliminaries

Throughout this paper all graphs are finite, undirected and simple (i.e. loop–free and without multiple edges).

Let $G = (V, E)$ be a graph with vertex set $V =: V(G)$ and edge set $E =: E(G)$. If no confusion can arise we write $n := |V|$ and $m := |E|$.

For an edge $\{u, v\}$ we write shortly uv. A vertex u is called a *neighbor* of v iff $uv \in E$. The set $N(v)$ of all neighbors of v is called the *neighborhood* of v. For a subset $U \subseteq V$ we define $N(U) := \bigcup_{u \in U} N(u)$. Further, we write $N[v] := N(v) \cup \{v\}$ and $N[U] := N(U) \cup U$.

A graph $G' = (V', E')$ is called a *subgraph* of G iff $V' \subseteq V$ and $E' \subseteq E$. G' is an *induced subgraph* iff G' is a subgraph of G and $E' = \mathfrak{P}_2(V') \cap E$.[1] We also say that G' *is induced by* V' and usually write $G(V')$ for G'. If $V' = (v_1, \ldots, v_k)$ is a k–tuple of vertices of V we analogously define $G(V') := G(\{v_1, \ldots, v_k\})$.

Let v be a vertex in V. For the graph $G(V \setminus \{v\})$ we write shortly $G - v$.

The *degree* of a vertex $v \in V$, i.e. the number of neighbors of v in G, is denoted by $\deg_G(v)$, the maximum degree of a vertex in G by $\triangle(G)$.

A sequence $P = (v_1, \ldots, v_k)$ of pairwise distinct vertices is a *path* in G iff for all $i \in \{1, \ldots, k-1\}$ holds $v_i v_{i+1} \in E$. The *length* of P is $k - 1$. P is *chordless* iff $|E(G(P))| = k - 1$.

A path $C = (v_1, \ldots, v_k)$ is a *cycle* in G iff $k > 2$ and $v_1 v_k \in E$. The *length* of C is k. C is *chordless* iff $|E(G(C))| = k$.

The *distance* $d(x, y)$ between two vertices x, $y \in V$ is the minimum length of a path between u and v, or ∞ if there is no such path. For $v \in V$ and $U \subseteq V$ we define $d(v, U) := \min_{u \in U} d(v, u)$.

The *k–th neighborhood* $N^k(v)$ of a vertex v of G is the set of all vertices of distance k to v, i.e.

$$N^k(v) := \{u \in V : d_G(u, v) = k\},$$

[1] $\mathfrak{P}_i(M)$ denotes the set of all i–element subsets of M.

whereas the *disk* of radius k centered at v is the set of all vertices of distance at most k to v:

$$D(v,k) := \{u \in V : d_G(u,v) \leq k\} \ = \bigcup_{i=0}^{k} N^i(v).$$

The *eccentricity* $e(v)$ of a vertex $v \in V$ is the maximum of $d(v,x)$ taken over all $x \in V$. The maximum over the eccentricities of all vertices of G is the *diameter* $\mathrm{diam}(G)$ of G.

G is *connected* iff for all u, $v \in V$ there is a path in G connecting u and v, otherwise we call G *disconnected*.

Let V' be a subset of V.

- V' is a *connected component* in G iff $G(V')$ is connected and for all $x \in V \setminus V'$ holds $G(V' \cup \{x\})$ is not connected.
- V' is a *clique (complete set)* in G iff for all u, $v \in V$, $u \neq v$ holds $uv \in E$.
- V' is a *stable set (independent set)* in G iff for all u, $v \in V$, $u \neq v$ holds $uv \notin E$.

G is *complete* iff $V(G)$ is a clique in G, G is *edgeless* iff $V(G)$ is a stable set in G.

A set $H \subseteq V$ is called *homogeneous set* iff any pair of vertices of H has the same neighborhood outside H:

$$N(u) \cap (V \setminus H) = N(v) \cap (V \setminus H) \qquad \text{for all } u, v \in H.$$

A homogeneous set H is *trivial* (resp. *nontrivial*) iff $|H| \in \{0, 1, |V|\}$ (resp. $1 < |H| < |V|$). A graph containing only trivial homogeneous sets is called *prime*.

Let $T = (V,E)$ be a tree, i.e. a connected cycle–free graph, rooted at w and let v some vertex of T. We denote by T_v the subtree of T rooted at v, i.e. T_v contains any vertex u such that v lies on the (unique) path connecting u and the root w.

Let $\mathcal{F}$ be a set of graphs. A graph G is called *$\mathcal{F}$-free* iff for all $F \in \mathcal{F}$ holds: F is not an induced subgraph of G. If $\mathcal{F} = \{F_1, \dots, F_\ell\}$ (resp. $\mathcal{F} = \{F\}$) is finite (resp. has exactly one element) we write $(F_1, \dots, F_\ell)$–free (resp. F–free) instead of $\mathcal{F}$-free.

Let $\mathcal{C}$ be a class of graphs. $\mathcal{C}$ is called *hereditary* iff for every $G \in \mathcal{C}$ and $U \subseteq V(G)$ holds: $G(U) \in \mathcal{C}$. Graph classes defined by forbidden subgraphs are examples for hereditary graph classes.

Let A, B be subsets of V. We write $A \circledone B$ iff for all $a \in A$, $b \in B$ holds $ab \in E$. We write $A \circledzero B$ iff for all $a \in A$, $b \in B$ holds $ab \notin E$.

For vertex–disjoint graphs $G = (V,E)$ and $G' = (V', E')$ we define

- $G \circledzero G' := (V \cup V', E \cup E')$, the *union of* G *and* G' (in the literature often the union is denoted by $G \cup G'$ or $G + G'$),
- $nG := \underbrace{G \circledzero \dots \circledzero G}_{n \text{ times}}$, $n \geq 1$ an arbitrary integer,
- $G \circledone G' := (V \cup V', E \cup E' \cup \{xy : x \in V, y \in V'\})$, the *join of* G *and* G'.

Some special graphs get a standard notation ($n \in \mathbb{N}$):

- P_n := graph of a chordless path on n vertices,
- C_n := graph of a chordless cycle on n vertices ($n > 2$),
- K_n := complete graph on n vertices.

By the above notion nK_1 is an edgeless graph on n vertices.

An induced connected subgraph H of G is an *isometric subgraph* of G iff the distance of any two vertices x, y in H equals their distance in G, i.e. $d_H(x, y) = d_G(x, y)$ for all $x, y \in V(H)$. Then, a connected graph G is *distance–hereditary* iff every induced connected subgraph of G is isometric ([69]).

In the 1980s certain characterizations of distance–hereditary graphs were given. A constructive generation of distance–hereditary graphs was presented in [24] via so–called one vertex extensions. In [77], analogous results were obtained including a linear time recognition algorithm which constructs a sequence of one vertex extensions.

Let $G' = (V', E')$ be a graph, $x' \in V'$ and $x \notin V'$. We extend the graph G' to G by adding vertex x and joining it to

- only x' — the *pendant vertex* operation PV,
- all neighbors of x' — the *false twin* operation FT,
- x' and all its neighbors — the *true twin* operation TT.

Then, a connected graph G with at least two vertices is distance–hereditary if and only if G can be obtained from an edge by a sequence of one vertex extensions PV, FT and TT.

2.1 Domination problems

The roots of domination lie in 1850 when NAUCK, GAUSS ET AL. studied the placement of chess pieces on a $n \times n$–board. They investigated the domination of all squares by a minimum number of pieces.

Later, the problem was formulated for graphs. A set $D \subseteq V$ is a *dominating set* in G if every vertex $v \in V \setminus D$ has at least one neighbor in D. A dominating set D is *minimum* if there is no dominating set D' with $|D'| < |D|$. The cardinality of a minimum dominating set in G is called *domination number* and denoted by $\gamma(G)$. The problem MINIMUM DOMINATING SET consists of computing a minimum dominating set for a given graph.

In the last years many variants of MINIMUM DOMINATING SET have been developed. Here, we want to mention four well–known general modifications:

- Additionally, we have given a radius function $r : V \to \mathbb{N}$. D is a *r-dominating set* if for every vertex $v \in V \setminus D$ there is at least one vertex $d \in D$ of distance at most $r(v)$ to v. If $r(v) = 1$ for all $v \in V$ we get the classical definition for dominating set. Note, that $r(v) = 0$ means that this vertex must belong to every r-dominating set.

- D is called a *perfect dominating set* if every vertex $v \in V \setminus D$ has exactly one neighbor in D.
- We study dominating sets D with additional properties:
  - $G(D)$ is connected
    $\rightsquigarrow$ MINIMUM CONNECTED DOMINATING SET,
  - D is an independent set in G
    $\rightsquigarrow$ MINIMUM INDEPENDENT DOMINATING SET,
  - D is *total* in G i.e. $G(D)$ contains no isolated vertices (= vertices of degree 0)
    $\rightsquigarrow$ MINIMUM TOTAL DOMINATING SET,
  - D is a clique
    $\rightsquigarrow$ MINIMUM DOMINATING CLIQUE.
- We consider (vertex) weighted versions. Let $w : V \to \mathbb{R}$ be a function assigning weights to vertices of G. A dominating set D is a *minimum weighted dominating set* if D is a dominating set in G and there is no dominating set D' in G with $\sum_{d \in D'} w(d) < \sum_{d \in D} w(d)$. If $w(v) = 1$ for all $v \in V$ the weighted version is the classical domination problem.

These modifications can also be combined. For example: D is a connected perfect dominating set if D is a perfect dominating set and $G(D)$ is connected.

Finally, we want to mention the problem (CARDINALITY) STEINER TREE. Let $T \subset V$ be a subset of vertices of V. D is called a *Steiner set* if D is a vertex set containing T such that $G(D)$ is connected. STEINER TREE consists of computing a Steiner set of minimum cardinality. This problem has a lot of applications in VLSI design and reconstruction of phylogenetic trees in biology.

2.1.1 Relationship between domination problems

In [124] the authors investigate the relationship between Steiner trees and connected dominating sets in chordal graphs. Using a convexity property they obtain

Theorem 2.1.1 ([124]) *The connected dominating set problem is polynomial for any class of chordal graphs for which the cardinality Steiner tree problem is polynomial. Moreover, the cardinality Steiner tree problem is $\mathbb{NP}$–complete for any subclass of chordal graphs for which the connected dominating set problem is $\mathbb{NP}$–complete.*

In [92] the authors investigate the relationship between MINIMUM DOMINATING SET and MINIMUM TOTAL DOMINATING SET. The work was motivated by the fact that in almost all of the known cases the two problems have the same complexity status. Looking in the Appendix one can find one exception: MINIMUM DOMINATING SET is $\mathbb{NP}$–complete on chordal–bipartite graphs whereas MINIMUM TOTAL DOMINATING SET on this class can be solved in polynomial time.

For a graph $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$, the *duplex graph* $D(G)$ has vertices $V' = V \cup \{u_1, \ldots, u_n\}$ and edges $E' = E \cup \{u_i u_j : v_i v_j \in E\} \cup \{u_i v_j : v_i v_j \in E\}$.

Theorem 2.1.2 ([92]) *Let $\mathfrak{G}$ be a graph class with $D(\mathfrak{G}) \subseteq \mathfrak{G}$. Then holds:*

1. *If there is an algorithm solving* MINIMUM DOMINATING SET *in time $O(f(n, m))$ then there is also an algorithm solving* MINIMUM TOTAL DOMINATING SET *in the same time bound.*

2. *If* MINIMUM TOTAL DOMINATING SET *is $\mathbb{NP}$–complete on $\mathfrak{G}$ then* MINIMUM DOMINATING SET *is $\mathbb{NP}$–complete on $\mathfrak{G}$, too.*

Since graph classes closed under adding false twins fulfill the property $D(\mathfrak{G}) \subseteq \mathfrak{G}$ of Theorem 2.1.2 ([92]) we can use this theorem for the following classes:

> AT–free graphs, bipartite graphs, chordal bipartite graphs, circle graphs, co–comparability graphs, comparability graphs, convex bipartite graphs, distance–hereditary graphs, dually chordal graphs, homogeneously orderable graphs, permutation graphs, k–polygon graphs.

Chapter 3

Special Graph Classes

Since nearly all domination problems are $\mathbb{NP}$–complete on general graphs it is interesting to investigate the complexity on special graph classes.

In the first section we consider graph classes of bounded clique width. For such classes under certain conditions there exists a unified approach to get efficient solutions for many graph theoretic problems. One condition is that the given problem can be expressed in terms of a logical expression in a so–called Monadic Second Order Logic. In particular we investigate which domination problems can be expressed in such a logic.

In section two we look at $\mathcal{F}$–free graphs where $\mathcal{F}$ is a set of graphs containing at most four vertices. For all of these classes we investigate if MINIMUM DOMINATING SET is $\mathbb{NP}$–complete or if it can be solved efficiently. For the case that only one graph is forbidden (i.e. H–free graphs) we give the complexity status for MINIMUM DOMINATING SET and MINIMUM CONNECTED DOMINATING SET for arbitrary graphs H (i.e. not only for the case $|V(H)| \leq 4$).

Next, we consider the relation of homogeneous sets to the minimum r–dominating set problem. We show that one can reduce a homogeneous set to one vertex or to a so–called meta–vertex consisting of two nonadjacent vertices with r–value one. This reduction together with modular decomposition leads to an $O(|V||E|)$ time algorithm for computing a minimum r–dominating set for graphs which can be generated from the one–vertex graph by a finite number of homogeneous extensions (substitution of an arbitrary graph into a vertex) and by attaching pendant vertices (leaves). For distance–hereditary graphs — a proper subclass of this graph class — we even get a linear time algorithm.

3.1 Classes of bounded clique width

Graphs of clique width p, $p \in \mathbb{N}$, were introduced by COURCELLE, ENGELFRIET and ROZENBERG (1993) in [32] as graphs which can be defined by p–expressions based on graph operations which use p vertex labels. In the first subsection we give the exact definition for clique width and some properties with respect to modular decomposition and graph complement.

Recently, the concept of clique width attracted much attention. In [40] the authors give an unified approach to get efficient solutions for many graph theoretic problems on graph classes of bounded clique width provided a p–expression is given. For this, the given problem must be expressible in terms of a logical expression in a so called Monadic Second Order Logic. This concept will be outlined in the second subsection. In particular, we investigate which domination problems can be expressed in Monadic Second Order Logic (see Theorem 3.1.3, Observation 3.1.4).

In [20] we investigate the structure and clique width of graph classes defined by exactly two forbidden four–vertex graphs. For nearly all combinations we prove that the clique width is bounded, or not; we list our results in the third subsection. To give the reader a feeling how to prove that a graph class has bounded or unbounded clique width we give one example for both cases.

Finally, we give an overview about the clique width of some important graph classes.

3.1.1 Definition and some properties

A *p–graph*, $p \in \mathbb{N}$, is a graph with vertex labels in $\{1, \dots, p\}$. Hereby, we force that every vertex has exactly one label. If G is a p–graph we denote with $L_i(G)$ the set of vertices with label i, $i = 1, \dots, p$.

We consider the following graph operations:

- (Generation of an one vertex p–graph)
  Let v be an object and $\ell \in \{1, \dots, p\}$. Then $\ell(v)$ denotes the p–graph $G = (\{v\}, \emptyset)$, $L_\ell(G) = \{v\}$.

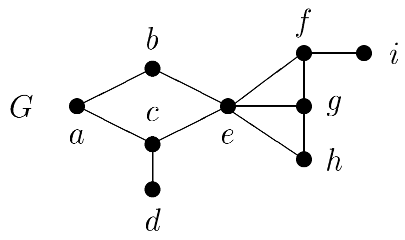
- (Disjoint union)
  Let G and H be p–graphs. Then, $G \oplus H$ denotes the disjoint union of G and H with vertex labels $L_i(G \oplus H) := L_i(G) \cup L_i(H)$, $i = 1, \dots, p$.

- (Join between vertices of different labels)
  Let G be a p–graph. $\eta_{i,j}(G)$ denotes the graph that we obtain from G if we add all edges between a vertex of label i and a vertex of label j and not changing any label.

- (Relabeling vertices of one label to another label)
  Let G be a p–graph. $\varrho_{i\to j}(G)$ denote the graph that we obtain from G if we change the label of all vertices with label i to j, i.e. $L_j(\varrho_{i\to j}(G)) = L_i(G) \cup L_j(G)$, $L_i(\varrho_{i\to j}(G)) = \emptyset$.

The *clique width* of a graph G, denoted by $cwd(G)$, is the minimum number of labels needed to generate G by using these four operations. The recursive generation sequence of building G using the above operations by using only p different labels is called a *p–expression* (see Figure 3.1 for an example).

If one want to show that a hereditary graph class $\mathcal{C}$ has bounded clique width it is enough to prove this property for all prime graphs contained in $\mathcal{C}$. Further, the property is invariant



$$
\begin{aligned}
G_1 &= \varrho_{1\to 3}(\eta_{1,2}(2(e) \oplus \varrho_{2\to 1}(\eta_{1,2}(\eta_{1,3}(1(f) \oplus 3(i)) \oplus 2(g) \oplus 1(h))))), \\
G &= \eta_{1,2}(\eta_{1,3}(3(d) + 1(c)) \oplus 1(b) \oplus 2(a) \oplus G_1), \\
&\Rightarrow cwd(G) = 3.
\end{aligned}
$$

Figure 3.1: A graph of clique width three.

for building complements, i.e. $\mathcal{C}$ has bounded clique width if and only if

$$\text{co–}\mathcal{C} := \{G : \overline{G} \in \mathcal{C}\}$$

has bounded clique width.

Theorem 3.1.1 ([40, 43]) *The clique width of a graph is the maximum of the clique width of its induced prime subgraphs, and the clique width of the complement graph $\overline{G}$ of G is at most twice the clique width of G.*

3.1.2 Monadic second–order logic and the class LinEMSOL($\tau_{1,p}$)

First, we define *first–order logic* (FOL) where quantification is allowed only over variables. Let $\mathcal{X} := \{x_i : i \in \mathbb{N}\}$ and $\mathcal{R} = \{R_i^j : i,\ j \in \mathbb{N}\}$. We call the elements of $\mathcal{X}$ *variables* and the elements of $\mathcal{R}$ *relation symbols*. Each relation symbol R_i^j has an *arity*, denoted by $\varrho(R_i^j)$, which is equal to the upper index j.

A *formula in first–order logic* is defined recursively as follows:

1. For $R \in \mathcal{R}$, $n = \varrho(R)$ and variables $x_1, \ldots, x_n$ let $R(x_1, \ldots, x_n)$ be a formula. These special formulas we call *atomic formulas*.

2. Let F, G be formulas and x be a variable. Then $F \wedge G$, $F \vee G$, $\neg F$, $\forall x F$, $\exists x F$ are formulas.

3. There are no more formulas.

Next, we have to give the semantics of our formulas. A *structure* for a formula F in first–order logic is a tuple $\mathcal{S} = (D_{\mathcal{S}}, i_{\mathcal{S}})$ consisting of a non–empty set $D_{\mathcal{S}}$, the *domain* of $\mathcal{S}$, and an interpretation $i_{\mathcal{S}}$ that assigns to every $R \in \mathcal{R}$ (if R occurs in F) a $\varrho(R)$–ary relation $R^{\mathcal{S}} := i_{\mathcal{S}}(R)$ on $D_{\mathcal{S}}$ and to every $x \in \mathcal{X}$ (if x occurs in F and x is in F unbounded by a quantor) an element $x^{\mathcal{S}} := i_{\mathcal{S}}(x)$ in $D_{\mathcal{S}}$. Finally, we have to assign to F a logical value $\mathcal{S}(F) \in \{0, 1\}$ under the structure $\mathcal{S}$. This is done recursively as follows:

1. If $F = R(x_1, \ldots, x_n)$ then $\mathcal{S}(F) = 1$ if and only if $((x_1)^{\mathcal{S}}, \ldots, (x_n)^{\mathcal{S}}) \in R^{\mathcal{S}}$.
2. If $F = \neg G$ then $\mathcal{S}(F) = 1 - \mathcal{S}(G)$.
3. If $F = G \wedge H$ then $\mathcal{S}(F) = \min(\mathcal{S}(G), \mathcal{S}(H))$.
4. If $F = G \vee H$ then $\mathcal{S}(F) = \max(\mathcal{S}(G), \mathcal{S}(H))$.
5. If $F = \forall x G$ then $\mathcal{S}(F) = \min_{d \in D_{\mathcal{S}}} \mathcal{S}_{[x/d]}(G)$.
6. If $F = \exists x G$ then $\mathcal{S}(F) = \max_{d \in D_{\mathcal{S}}} \mathcal{S}_{[x/d]}(G)$.

Hereby, $\mathcal{S}_{[x/d]}$ is the extension of $\mathcal{S}$ by $x \mapsto d$. So, first–order logic coincides with the usual first order predicate logic without function symbols (see for instance [115]).

Next, *second–order logic* (SOL) is an extension of first–order logic in terms of that quantification is allowed over relation symbols, too. Syntax and semantics are modified naturally — we omit the details.

Now, we are able to say what *monadic second–order logic* (MSOL) is. In this special kind of second–order logic quantification is allowed only over variables and relation symbols of arity one. If R is a relation symbol of arity one and x is a variable then we write $x \in R$ instead of $R(x)$.

For expressing properties of (labeled) graphs we use special formulas and structures. Let $\mathcal{L} \in \{\text{FOL}, \text{SOL}, \text{MSOL}\}$ and p be an integer. With $\mathcal{F}_{1,p}$ we denote the set of all formulas F in logic $\mathcal{L}$ that fulfill the following conditions:

- F contains only the following free relation symbols: edg, eq, $L_1, \ldots, L_p$.
- edg and eq have arity two. We write $(x, y) \in edg$ (resp. $x = y$) for $edg(x, y)$ (resp. $eq(x, y)$).
- $L_1, \ldots, L_p$ are unary.

Let $G = (V, E)$ be a p–graph and $F \in \mathcal{F}_{1,p}$. We define the structure $\mathcal{S}^G_{1,p}$ as follows:

- $D_{\mathcal{S}^G_{1,p}} := V$,
- for all v, $w \in V$ holds $(v, w) \in edg^{\mathcal{S}^G_{1,p}}$ if and only if $vw \in E$,[1]
- for all v, $w \in V$ holds $(v, w) \in eq^{\mathcal{S}^G_{1,p}}$ if and only if $v = w$,
- for all $v \in V$ and $i \in \{1, \ldots, p\}$ holds $v \in L_i^{\mathcal{S}^G_{1,p}}$ if and only if vertex v has label i.

We use $\tau_{1,p} = (\mathcal{F}_{1,p}, \{\mathcal{S}^G_{1,p} : G \text{ a } p\text{–graph}\})$ for expressing properties of labeled graphs. The index ”1” denotes that this is one possibility — see [31, 40] for other variants.

Let π be a property for graphs. We say that a formula $F \in \mathcal{F}_{1,p}$ *expresses* π if the following property is fulfilled:

[1]Note, that we only consider undirected graphs.

For all p–graphs G holds: $\mathcal{S}^G_{1,p}(F) = 1$ if and only if G fulfills property π.

In the following we give some examples for such expressions. We use the following abbreviations:

$$x \to y := \neg x \vee y, \qquad x \neq y := \neg(x = y).$$

Theorem 3.1.2 *The following graph properties can be expressed by a formula in $MSOL$:*

1. *G is complete,*
2. *G is edgeless,*
3. *G is total, i.e. G has no isolated vertices,*
4. *G is connected,*
5. *G is cycle–free.*

Proof. The following formulas express the given properties:

1. $F_1 = \forall x \forall y\ (x \neq y) \to (x, y) \in edg$,
2. $F_2 = \forall x \forall y\ (x \neq y) \to \neg\ (x, y) \in edg$,
3. $F_3 = \forall x \exists y\ (x \neq y) \wedge (x, y) \in edg$,
4. $F_4 = \forall X\{(\exists u\ u \in X) \wedge [\forall v \forall w\ (v \in X \wedge (v, w) \in edg) \to w \in X]\} \to (\forall x\ x \in X)$,
5. We use the following characterization: A graph G is cycle–free iff there is no vertex $x \in V$ such that two different neighbors y, $z \in N(x)$ of x are in the same connected component in $G - x$.

$$\begin{aligned} F_5 \ = \ & \forall x \forall y \forall z\ ((x, y) \in edg \wedge (x, z) \in edg \wedge y \neq z) \\ & \to \Big\{ \exists U\ \neg\ x \in U \wedge \neg\ z \in U \wedge y \in U \wedge \\ & \qquad \forall u \forall v [(u \in U \wedge x \neq v \wedge (u, v) \in edg) \to v \in U] \Big\}. \end{aligned}$$

□

Analogously we are able to express optimization problems. We call an optimization problem π a LinEMSOL($\tau_{1,p}$) problem if and only if for every instance G of π the following properties are fulfilled:

- there is a MSOL–formula $F := F(X_1, \ldots, X_\ell)$ where $X_1, \ldots, X_\ell$ are free relation symbols of arity one and it holds: $(T_1, \ldots, T_\ell)$ is a solution for (π, G) iff $\mathcal{S}^G_{1,p\ [X_1/T_1]\cdots[X_\ell/T_\ell]}(F) = 1$,

- there are evaluation functions $f_1, \ldots, f_m$ associating integer values to the vertices of G, integers $a_{i,j}$, $1 \leq i \leq \ell$, $1 \leq j \leq m$ and $\mathrm{opt} \in \{\min, \max\}$ such that the following value is the optimum solution for the instance G of π:

$$\mathrm{opt}\left\{\sum_{i=1}^{\ell}\sum_{j=1}^{m} a_{ij} \sum_{a \in T_i} f_j(a) : (T_1, \ldots, T_\ell) \text{ is a solution for } (\pi, G)\right\}.$$

Theorem 3.1.3 *The following optimization problems are for every fixed integer k in LinEMSOL($\tau_{1,p}$):*

1. MINIMUM WEIGHTED (k–)DOMINATING SET,
2. MINIMUM WEIGHTED PERFECT DOMINATING SET,
3. MINIMUM WEIGHTED CONNECTED (k–)DOMINATING SET,
4. MINIMUM WEIGHTED CONNECTED PERFECT DOMINATING SET,
5. MINIMUM WEIGHTED INDEPENDENT (k–)DOMINATING SET,
6. MINIMUM WEIGHTED INDEPENDENT PERFECT DOMINATING SET,
7. MINIMUM WEIGHTED TOTAL (k–)DOMINATING SET,
8. MINIMUM WEIGHTED TOTAL PERFECT DOMINATING SET,
9. MINIMUM WEIGHTED (k–)DOMINATING CLIQUE,
10. STEINER TREE.

Proof. The following formula expresses that a set is a dominating set in G:

$$F_1(X) := \forall x\ x \in X \vee \exists y\ [y \in X \wedge (x, y) \in edg].$$

It is easy to see that for fixed k one can also express that a set is a k–dominating set. Therefore, with $\ell = m = a_{11} = 1$ and f_1 the given function associating weights to the vertices one can see that MINIMUM WEIGHTED (k–)DOMINATING SET is an element in LinEMSOL($\tau_{1,p}$).

The following formula expresses that a set is a perfect dominating set in G:

$$\begin{aligned} F_2(X) \quad := \quad & \forall x\ x \in X \vee \Big\{\exists y\ y \in X \wedge (x, y) \in edg \\ & \qquad\qquad \wedge \forall z\ [(z \in X \wedge (x, z) \in edg) \rightarrow y = z]\Big\}. \end{aligned}$$

According to Theorem 3.1.2 we can express the following additional properties of the (perfect) dominating set D to the formula implying the membership of the remaining problems to LinEMSOL($\tau_{1,p}$): $G(D)$ is connected, resp. edgeless, resp. complete, resp. total, resp. cycle–free. □

Note, that in Theorem 3.1.3 the condition that k is fixed is sufficient since in $\tau_{1,p}$ no labels larger than p are allowed implying that we cannot build an MSOL–formula for the solutions.

Observation 3.1.4 *For unfixed integer k the problem* MINIMUM k–DOMINATING SET *(and thus the variants of this problem) is not a problem in LinEMSOL($\tau_{1,p}$). Furthermore,* MINIMUM r–DOMINATING SET *(and the variants of this problem) is not a problem in LinEMSOL($\tau_{1,p}$).*

The following result gives a relation of classes of bounded clique width to optimization problems in LinEMSOL($\tau_{1,p}$):

Theorem 3.1.5 ([40]) *Let $\mathcal{G}$ be a graph class of clique width at most p such that there is an $O(f(|E|,|V|))$ algorithm constructing a p–expression for every graph $G \in \mathcal{G}$. Then, every LinEMSOL($\tau_{1,p}$) problem on $\mathcal{G}$ can be solved in $O(f(|E|,|V|))$ time. A corresponding algorithm can be constructed from the logical formula describing the problem and the parsing algorithm for the class.*

So, if the clique width of a class $\mathcal{C}$ of graphs is bounded by p and a p–expression can be computed for all $G \in \mathcal{C}$ efficiently then every optimization problem in LinEMSOL($\tau_{1,p}$) can be solved on $\mathcal{C}$ efficiently. This result gives a unified approach solving many graph theoretic problems on classes of bounded clique width.

For domination it is an interesting open question if there is an extension $\mathcal{X}$ of LinEMSOL($\tau_{1,p}$) such that k–Domination for unfixed k and r–Domination can be expressed in $\mathcal{X}$ without destroying the property of Theorem 3.1.5.

3.1.3 Graph classes defined by two forbidden four–vertex graphs

In [20] we investigate the structure and clique width of graph classes defined by two forbidden four–vertex graphs. For nearly all combinations we determine whether the clique width is bounded or not; in Figure 3.2 our results are listed. By Theorem 3.1.1 it remains to investigate the status of clique width for the following two classes:

1. $(K_4, 2K_2)$-free graphs,
2. $(K_4$,co-diamond)-free graphs.

To give the reader a feeling how to prove that a graph class has bounded or unbounded clique width we give one example for both cases.

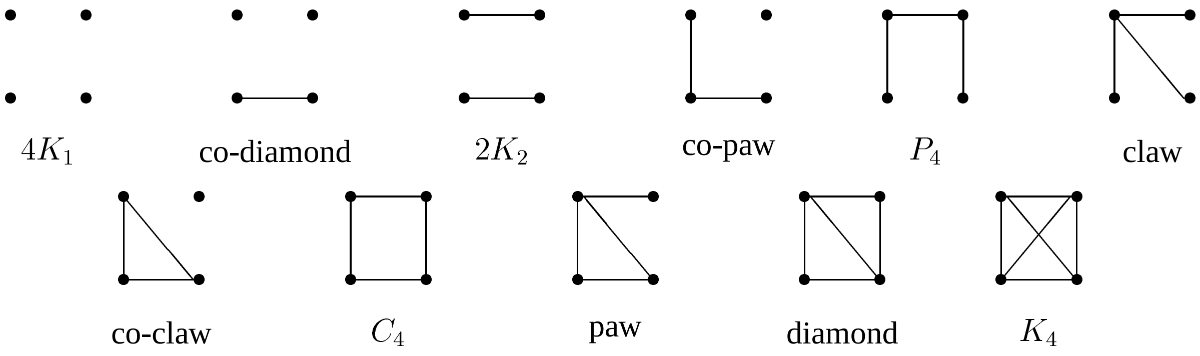
First, we want to consider a graph class of bounded clique width.

Theorem 3.1.6 ([20]) *If G is a prime (K_4,co-paw)-free graph then G has at most nine vertices.*

Proof. Assume first that G contains a K_3 a, b, c.

We call a vertex $v \in V \setminus \{a, b, c\}$ an i–vertex, $i \in \{0, 1, 2, 3\}$, if $|N(v) \cap \{a, b, c\}| = i$. Let N_a (resp. N_b, N_c) denote the set of 1-vertices adjacent to a (resp. b, c), let $N_{a,b}$ (resp. $N_{a,c}$, $N_{b,c}$) denote the set of 2-vertices adjacent to a, b (resp. a, c, b, c) and let R denote the set of 0-vertices with respect to a, b, c.[2]

[2]Note: Since G is K_4–free G contains no 3–vertices.



G_i / G_j	$4K_1$	co-diamond	$2K_2$	co-paw	claw	P_4	co-claw	C_4	paw	diamond	K_4
$4K_1$	∞	∞	∞	∞	∞	≤ 2	∞	?	bcw	?	bcw
co-diamond	∞	∞	∞	∞	∞	≤ 2	∞	bcw	bcw	bcw	?
$2K_2$	∞	∞	∞	∞	∞	≤ 2	∞	∞	bcw	bcw	?
co-paw	∞	∞	∞	∞	∞	≤ 2	bcw	bcw	bcw	bcw	bcw
claw	∞	∞	∞	∞	∞	≤ 2	bcw	∞	bcw	∞	∞
P_4	≤ 2	≤ 2	≤ 2	≤ 2	≤ 2	≤ 2	≤ 2	≤ 2	≤ 2	≤ 2	≤ 2
co-claw	∞	∞	∞	bcw	bcw	≤ 2	∞	∞	∞	∞	∞
C_4	?	bcw	∞	bcw	∞	≤ 2	∞	∞	∞	∞	∞
paw	bcw	bcw	bcw	bcw	bcw	≤ 2	∞	∞	∞	∞	∞
diamond	?	bcw	bcw	bcw	∞	≤ 2	∞	∞	∞	∞	∞
K_4	bcw	?	?	bcw	∞	≤ 2	∞	∞	∞	∞	∞

Figure 3.2: Clique width of (G_i, G_j)-free graphs. ’bcw’ (resp. ’∞’) means bounded (resp. unbounded) clique width. For the four cases marked with ’?’ it is unknown if the clique width is bounded.

Note, that since G is K_4-free, $N_{a,b}$, $N_{a,c}$ and $N_{b,c}$ are stable sets. Moreover, since G is co-paw-free, every vertex $v \notin \{a, b, c\}$ has distance at most 2 to a, b, c.

Claim 1. R and $N_{a,b}$, $N_{a,c}$ and $N_{b,c}$ are homogeneous sets and thus have at most one vertex.

We first show that R is a homogeneous set. Assume not, and let $x, y \in R$ distinguished by $z \notin R$. If $xy \in E$ then, since $z \notin \{a, b, c\}$ and G is K_4-free, z is nonadjacent to one of a, b, c, say c but then $xyzc$ induce a co-paw. If $xy \notin E$ then, since z has a neighbor in $\{a, b, c\}$, say a, the vertices x, z, a, y induce a co-paw — a contradiction in both cases.

Now assume that $N_{a,b}$ is no homogeneous set. Let $x, y \in N_{a,b}$ distinguished by $z \notin N_{a,b}$. If $z \notin R$ then z is adjacent to $\{a, b, c\}$. Since $z \notin N_{a,b}$, z is adjacent to c but now $cxzy$ induce a co-paw. If $z \in R$ then $ybcz$ induce a co-paw — a contradiction in both cases. The proof is similar for $N_{a,c}$ and $N_{b,c}$.

Claim 2. N_a, N_b and N_c are connected by a join.

Let $x \in N_a$ and $y \in N_b$. Since $ybcx$ is no co-paw, $xy \in E$. The other proofs are similar.

Claim 3. R is adjacent to every vertex in $N(a, b, c)$.

If not, G will contain a co-paw.

Claim 4. N_x and $N_{x,y}$, x, $y \in \{a, b, c\}$, $x \neq y$, are connected by a join.

Let $s \in N_x$ and $t \in N_{x,y}$. Say z is the remaining vertex in $\{a, b, c\}$, i.e. $\{x, y, z\} = \{a, b, c\}$. Since $styz$ is no co-paw, $st \in E$.

Claim 5. For all $x \in \{a, b, c\}$ holds $|N_x| \leq 1$ or $|N_x| = 2$ and N_x is an edge.

Say $x = a$. Since N_a is (K_3, P_3)-free N_a induces a $\ell K_2 \textcircled{0} mK_1$ for suitable ℓ, $m \in \mathbb{N}_0$. Assume $|N_a| \geq 2$. Since N_a is no homogeneous set and by Claim 2, 3 and 4 two vertices of N_a can only be distinguished by vertices from $N_{b,c}$ we conclude $N_{b,c} = \{d\}$ for some d. Assume p, $q \in N_a$, $pq \notin E$, can be distinguished by d. Then, $pdcq$ is a co-paw — a contradiction. Thus, N_a must be an edge.

Altogether G can contain at most nine vertices.

Now, assume that G is K_3-free. We consider two cases:

Case 1. G is P_3-free. Then $G = \ell K_1 \textcircled{0} mK_2$ for some $\ell, m \in \mathbb{N}_0$ and thus $|V(G)| \leq 2$.

Case 2. G contains a P_3 (a, b, c). Since G is co-paw-free every vertex is adjacent to some vertices of the P_3. It is easy to see that N_a, N_b, N_c, $N_{a,c}$ are pairwise connected by a join implying $|V(G)| \leq 5$. □

Next, we want to consider a graph class with unbounded clique width:

Theorem 3.1.7 ([20]) *(K_4,diamond,C_4,claw)–free graphs are not of bounded clique width.*

The proof of Theorem 3.1.7 is similar to the proofs given in [105]. We have to show that (K_4,diamond,C_4,claw)–free graphs contain graphs with 'good' grid structure that have unbounded clique width.

First, we need some definitions: Let $G = (V, E)$ be a graph and $V = V_{red} \uplus V_{blue}$ be a partition of V into red and blue vertices. As follows we define an equivalence relation $\sim_{red}$ on the vertices contained in V_{red}:

$u \sim_{red} v$ if and only if there is no blue vertex which distinguishes u and v.

The *2-color-width* of G, denoted as $2colw(G)$, is defined as the smallest number $\ell \in \mathbb{N}$, such that there is a partition $V = V_{red} \uplus V_{blue}$ of the vertices of G into two sets such that $|V|/3 \leq |V_{red}| \leq 2|V|/3$ and $\sim_{red}$ has exactly ℓ equivalence classes. The following theorem gives the relation between 2-color-width and clique width:

Theorem 3.1.8 ([105]) *For every graph G, if $cwd(G) \leq k$ then $2colw(G) \leq k$.*

By this Theorem graphs with unbounded 2-color-width have unbounded clique width, too.

Now, we define a set of special (K_4,diamond,C_4,claw)–free graphs which have a 'good' grid structure. Let G_n be a $n \times n$ grid, $n \in \mathbb{N}$. For every vertex v occurring in G_n at column i and row j we write $col(v) := i$ and $row(v) := j$. We construct a graph L_n from G_n by the following steps:

1. Replace every edge of G_n by a simple path of length three, introducing two new vertices which are the internal vertices of the path. Let G'_n denote the resulting graph.

2. Let v be a vertex of degree three in G'_n. Then, there are exactly two neighbors u, w of v which are in the same row or column in G'_n. Add the edge uw. This step is repeated for all vertices of degree three in G'_n. Let G''_n denote the resulting graph.

3. Let v be a vertex of degree four in G''_n, and let $u_1, \ldots, u_4$ be the four neighbors of v in G''_n in clockwise order starting from west (i.e. u_1, u_3 (resp. u_2, u_4) are in the same row (resp. column) with v, $col(u_1) < col(u_3)$ and $row(u_2) < row(u_2)$). Add the edges u_1u_2 and u_3u_4. This step is repeated for all vertices of degree four in G''_n. L_n is the resulting graph.

As an example in Figure 3.3 the graph L_4 is shown. Clearly, every graph L_n, $n \in \mathbb{N}$, is (K_4, diamond, C_4, claw)–free. It remains to show:

Theorem 3.1.9 *For every $n \in \mathbb{N}$ holds $2colw(L_n) \geq n/3$.*

Proof. First note that $L_n =: (V, E)$ contains $5n^2 - 4n$ vertices. Suppose that $2colw(L_n) < n/3$. Then there exists a partition $V = V_{red} \uplus V_{blue}$ of V such that

$$\frac{5}{3}n^2 - \frac{4}{3}n \leq |V_{red}| \leq \frac{10}{3}n^2 - \frac{8}{3}n$$

and $\sim_{red}$ contains less than $n/3$ equivalence classes. In the following we suppose the vertices of L_n are arranged in $3n - 2$ rows and $3n - 2$ columns. We say a vertex v is a *red row-alternating vertex* if $v \in V_{red}$ and v has a neighbor $u \in V_{blue}$ in the same row. Analogously, we call a vertex v a *red column-alternating vertex* if $v \in V_{red}$ and v has a neighbor $u \in V_{blue}$ in the same column.

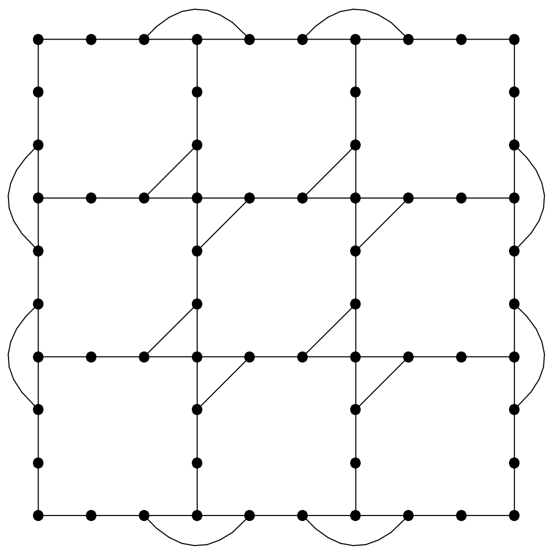


Figure 3.3: The graph L_4.

Case 1. For every $i \in \{1, 4, 7, \ldots, 3n-2\}$ there exists a red row-alternating vertex v_i in row i.
Then, it is easy to see that $v_1, v_4, v_7, \ldots, v_{3n-2}$ are n vertices which belong to n different equivalence classes of $\sim_{red}$. Therefore this case cannot occur.

Case 2. For some $i \in \{1, 4, 7, \ldots, 3n-2\}$ row i contains no red row-alternating vertex.

Case 2a. All vertices in row i are red. We construct a set Q of red vertices according to the following algorithm:

Algorithm CONSTRUCT Q

$j := 1$; $Q := \emptyset$;
repeat
    if not all vertices in column j are red
        then let v be any red column-alternating vertex in column j;
            $Q := Q \cup \{v\}$;
    else $\{$all vertices in column j are red$\}$
        check whether all vertices in columns $j+1$, $j+2$ are red;
        if 'no' **then** let v be any red vertex occurring in column $j+m$ which has a blue neighbor in column $j+m+1$, $m \in \{0, 1\}$;
        $Q := Q \cup \{v\}$;
    $j := j + 3$;
until $j = 3n - 2$;

graph class	clique width	where
$(q, q-3)$ graphs, $q \geq 7$	$\leq q$	[105]
$(q, q-4)$ graphs, $q \geq 4$	$\leq q$	[40]
$(q, q-1)$ graphs, $q \geq 4$	unbounded	[105]
$(6, 3)$ graphs, $(7, 5)$ graphs	unbounded	[105]
bipartite permutation	unbounded	[22]
cographs	≤ 2	folk
distance–hereditary	≤ 3	[68]
partial k–trees	$\leq 2^{k+1} + 1$	[43]
permutation	unbounded	[68]
split	unbounded	[105]
square grids	unbounded	[68]
trees	≤ 3	[43]
unit interval	unbounded	[68]

Table 3.1: Some graph classes with bounded an unbounded clique width.

Since by construction all vertices of Q occur in different equivalence classes of $\sim_{red}$ we conclude $|Q| < n/3$. Therefore, there are at least $2n/3$ columns j in $\{1, 4, 7, \ldots, 3n-2\}$ such that all vertices at columns j and (if $j \neq 3n-2$) $j+1$ and $j+2$ are red. Counting the number of red vertices in these columns and the red vertices in row i, we get

$$|V_{red}| > \frac{2}{3}n((3n-3) + (2n-2)) - (2n-2) + (3n-2) = \frac{10}{3}n^2 - \frac{7}{3}n$$

and conclude $|V_{red}| > 10/3n^2 - 8/3n$, therefore this case cannot occur.

Case 2b. All vertices in row i are blue. This case is handled similar to Case 2a above. Now, we reach a contradiction. □

3.1.4 Overview

In Table 3.1 we collect a list of graph classes of bounded and unbounded clique width. See [11] for a survey on the clique width of graph classes defined by three forbidden P_4 extensions.

3.2 Graph classes defined by small forbidden subgraphs

Many hereditary graph classes can be characterized by forbidden induced subgraphs or, if not, at least a list of small forbidden subgraphs is known (see [23] for an overview of such classes). In this section we consider $\mathcal{F}$–free graphs where $\mathcal{F}$ is a set of graphs containing at most four vertices. For all of these classes we investigate if MINIMUM DOMINATING SET is $\mathbb{NP}$–complete or if it can be solved efficiently. For the case that only one graph is forbidden (i.e. H–free graphs) we give the complexity status for arbitrary graphs H (i.e. not only for the case $|V(H)| \leq 4$).

3.2.1 H–free graphs

At first we consider graph classes which can be characterized by one forbidden induced subgraph.

Theorem 3.2.1 *Let H be a graph. If $H = P_k \mathbin{\textcircled{0}} \ell K_1$ for some $k \in \{1, 2, 3, 4\}$, $\ell \in \mathbb{N}_0$ then* MINIMUM DOMINATING SET *can be solved in $O(n^{\ell+2}(n + m))$ time on H–free graphs. Otherwise, the problem is $\mathbb{NP}$–complete.*

Proof. First let $H = P_k \mathbin{\textcircled{0}} \ell K_1$ for some $k \in \{1, 2, 3, 4\}$, $\ell \in \mathbb{N}_0$ and $G = (V, E)$ be an arbitrary H–free graph. We have to show that in this case MINIMUM DOMINATING SET can be solved efficiently. If G is P_4-free then a minimum dominating set can be computed in linear time ([45]). Thus, let G contain P_4 (a, b, c, d). Consider the following algorithm:

Algorithm DOMP4

(1) $D := \{a, b, c, d\}$;
(2) **while** D is **not** a dominating set in G **do**
(3)   **begin**
(4)   Let x be a vertex in $V \setminus N[D]$;
(5) $D := D \cup \{x\}$;
(6)   **end**;

By construction $D \setminus \{a, b, c, d\}$ is stable and every vertex is nonadjacent to any vertex of the P_4 (a, b, c, d). Since G is H–free the while-loop starting in line (2) passes through at most $\ell - 1$ times implying $|D| \leq \ell + 3$. Therefore $\gamma(G) \leq \ell + 3$ and a minimum dominating set can be computed in $O(n^{\ell+2}(n + m))$ time.

Now, let $H \neq P_k \mathbin{\textcircled{0}} \ell K_1$ for any $k \in \{1, 2, 3, 4\}$, $\ell \in \mathbb{N}_0$ and $G = (V, E)$ be an arbitrary H–free graph. We consider three cases:

Case 1. H contains a C_3.
Then bipartite graphs (= $\{C_3, C_5, C_7, \ldots\}$–free graphs) are H-free implying the $\mathbb{NP}$–completeness of MINIMUM DOMINATING SET on H–free graphs ([49]).

Case 2. H contains a C_4 or C_5 or $2K_2$.
Then split graphs (= $(C_4, C_5, 2K_2)$–free graphs) are H-free implying the $\mathbb{NP}$–completeness of MINIMUM DOMINATING SET on H–free graphs ([45]).

Case 3. H contains no C_3, C_4, C_5 and $2K_2$.
Since every cycle C_m, $m \geq 6$, contains a $2K_2$ H is a forest. Further, H contains at most one nontrivial connected component, otherwise H contains a $2K_2$. On assumption this component is not a P_m, $m \in \{1,2,3,4\}$. Hence, H must contain a claw implying that claw–free graphs are H–free. Thus, MINIMUM DOMINATING SET is $\mathbb{NP}$–complete for H–free graphs ([74]). □

Theorem 3.2.2 *Let H be a graph. If $H = P_k \textcircled{0} \ell K_1$ for some $k \in \{1,2,3,4\}$, $\ell \in \mathbb{N}_0$ then* MINIMUM CONNECTED DOMINATING SET *can be solved in $O(n^{2\ell+1}(n+m))$ time on H–free graphs. Otherwise, the problem is $\mathbb{NP}$–complete.*

Proof. The proof is similar to the proof of Theorem 3.2.1. For the case $H = P_k \textcircled{0} \ell K_1$ for some $k \in \{1,2,3,4\}$, $\ell \in \mathbb{N}_0$ and $G = (V,E)$ be an arbitrary H–free graph we only have to exchange Algorithm DOMP4 as follows:

Algorithm DOMP4CONN

(1) $D := \{a,b,c,d\}$;
(2) **while** D is **not** a dominating set in G **do**
(3) **begin**
(4) Let y be a vertex in $V \setminus N[D]$ such that $d(y,D) = 2$;
(5) Let $x \in N(D)$ with $xy \in E$;
(6) $D := D \cup \{x,y\}$;
(7) **end**;

Let Y be the set of all y-vertices taken in (4). By construction Y is stable and every vertex of Y is nonadjacent to any vertex of $\{a,b,c,d\}$. Since G is H–free the while-loop starting in line (2) passes through at most $\ell - 1$ times implying $|D| \leq 4 + 2(\ell-1) = 2\ell + 2$. Since D is connected $\gamma_c(G) \leq 2\ell + 2$ and a minimum connected dominating set can be computed in $O(n^{2\ell+1}(n+m))$ time.

The $\mathbb{NP}$-completeness proof for the case that $H \neq P_k \textcircled{0} \ell K_1$ for any $k \in \{1,2,3,4\}$, $\ell \in \mathbb{N}_0$ is analogously to the proof of Theorem 3.2.1 using [67, 97, 111, 124]. □

3.2.2 (H_1, H_2)–free graphs

Next, we investigate graph classes which can be characterized by exactly two forbidden subgraphs with at most four vertices. In Figure 3.4, 3.5 and 3.6 all graphs with at least two and at most four vertices are shown.

Immediately, by Theorem 3.2.1 we conclude

Observation 3.2.3 *The following propositions are true:*

Figure 3.4: Graphs containing exactly two nodes.

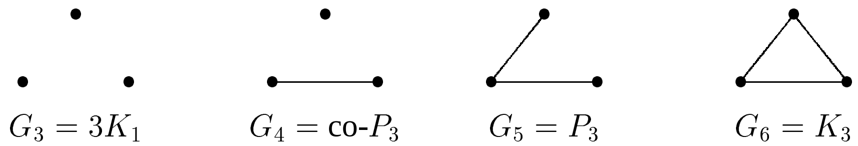


Figure 3.5: Graphs containing exactly three nodes.

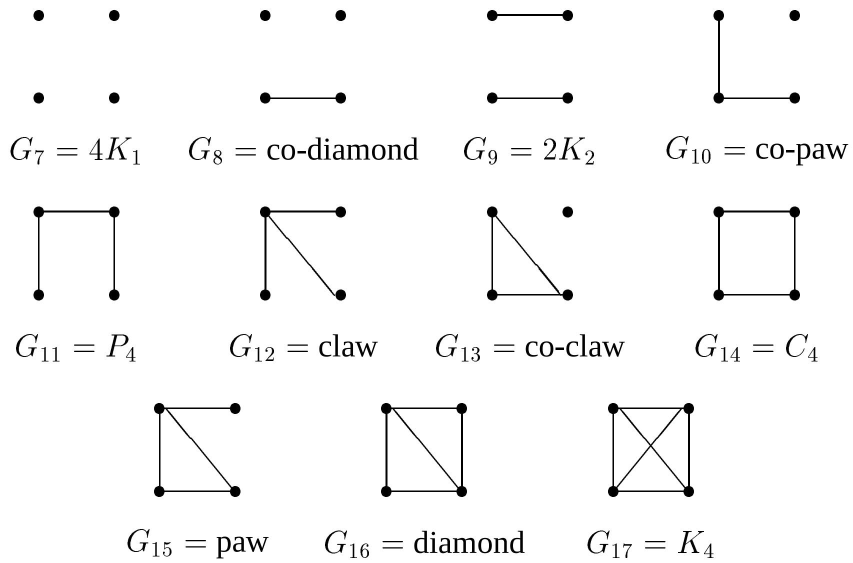


Figure 3.6: Graphs containing exactly four nodes.

1. *If* $i \in \{1,2,3,4,5,7,8,10,11\}$ MINIMUM DOMINATING SET *(*MINIMUM CONNECTED DOMINATING SET*) can be solved in polynomial time on* G_i*-free graphs.*

2. *If* $i \in \{6,9,12,13,14,15,16,17\}$ MINIMUM DOMINATING SET *(*MINIMUM CONNECTED DOMINATING SET*) is* $\mathbb{NP}$*-complete on* G_i*-free graphs.*

By this observation we need only to investigate (G_i, G_j)-free graphs for $i,\ j \in \{6,9,12,13,14,15,16,17\}$, $i \neq j$.

Immediately we get some $\mathbb{NP}$-complete cases by

Observation 3.2.4 *The following propositions are true:*

1. (G_i, G_i)*-free =* G_i*-free for all* $i \in \{1,\ldots,17\}$*.*

2. G_6*-free =* (G_6, G_i)*-free for* $i = 13, 15, 16, 17$*.*

3. *If* $(i,j) \in \{(13,15),(13,16),(13,17),(15,16),(15,17),(16,17)\}$ *then* G_6*-free* $\subset$ (G_i, G_j)*-free.*

4. *Split graphs are* (G_9, G_{14})*-free.*

Theorem 3.2.5 ([132]) *For every* $k \in \mathbb{N}$ MINIMUM DOMINATING SET *is* $\mathbb{NP}$*-complete on* $(C_4, \ldots, C_{2k+2})$*–free planar bipartite graphs with maximum degree three.*

Corollary 3.2.6 MINIMUM DOMINATING SET *is* $\mathbb{NP}$*-complete on* (K_3, C_4)*–free =* $(K_3, C_4,$ *co-claw*, *paw*, *diamond*, $K_4)$*-free graphs.*

By the results from subsection 3.1.3 we immediately get some cases where the clique width is bounded and thus MINIMUM DOMINATING SET can be solved in linear time:

Theorem 3.2.7 ([20]) *For the following parameters* (i,j) (G_i, G_j)*–free graphs have bounded clique width and a* p*–expression can be computed in linear time: (6,9), (6,12), (9,15), (9,16), (12,13), (12,15).*

Corollary 3.2.8 MINIMUM DOMINATING SET *can be solved on (*$2K_2$*, co–claw)–free graphs in* $O(n^3)$ *time.*

Proof. Let G be a ($2K_2$, co–claw)–free graph. If G is K_3–free then by Theorem 3.2.7 MINIMUM DOMINATING SET can be solved in linear time. If G contains a K_3 this K_3 dominates the whole graph since G is co–claw free. Thus, $\gamma(G) \leq 3$ and a minimum dominating set can be computed in $O(n^3)$ time. □

In [27] the authors show that every connected P_5–free graph has a dominating clique or a dominating P_3. Using this result they get a polynomial time algorithm for MINIMUM DOMINATING SET on $(2K_2, K_4)$–free graphs.

Theorem 3.2.9 ([27]) MINIMUM DOMINATING SET *can be solved on* $(2K_2, K_4)$*–free graphs in* $O(n^3)$ *time.*

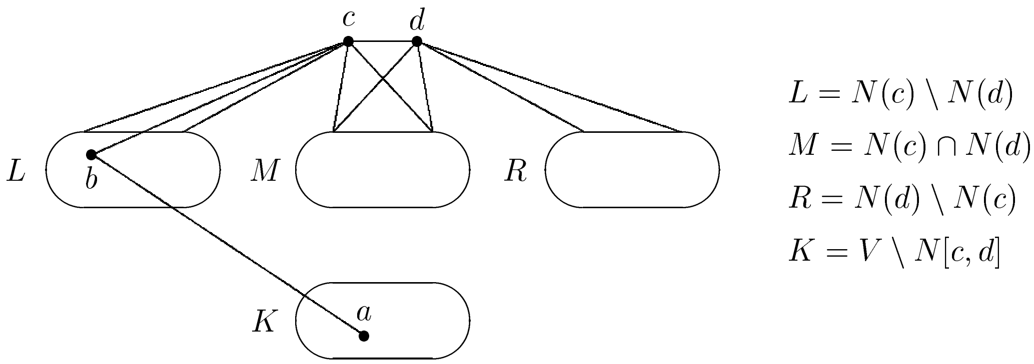


Figure 3.7: Hanging of the edge cd.

Results on (claw, H)-free graphs

By Theorem 3.2.7 MINIMUM DOMINATING SET can be solved on (claw,K_3)–free and (claw, co-claw)–free graphs in linear time since these classes have bounded clique width and a p–expression can be computed in linear time. For (claw,K_3)–free one can see this also directly:

Observation 3.2.10 *Every (claw,K_3)–free graph is a disjoint union of induced paths and induced cycles greater than three.*

Next, we want consider (claw,$2K_2$)-free graphs.

Theorem 3.2.11 *For (claw,$2K_2$)-free graphs* MINIMUM DOMINATING SET *can be solved in linear time.*

Proof. Let $G = (V, E)$ be a (claw,$2K_2$)-free graph. If G is P_4-free then a minimum dominating set can be computed in linear time ([45]). Therefore, let G contain a P_4 (a, b, c, d). We consider the hanging of the edge cd (see Figure 3.7). The defined sets have the following properties:

(P1) K is stable since G is $2K_2$-free.

(P2) L and R are complete since G is claw-free.

(P3) $|N(x) \cap K| \leq 1$ for all $x \in L \cup M \cup R$ since (P1) and G is claw-free.

(P4) $b \in L$, $a \in K$.

We consider two cases:

Case 1. $|K| \geq 2$.
Let $K = \{k_1, \ldots, k_\ell\}$. We partition $L \cup M \cup R$ according to their neighborhood in K:

$$L \cup M \cup R = V_1 \uplus \ldots \uplus V_\ell \uplus V_{\ell+1},$$
$$N(k_i) = V_i \quad (i = 1, \ldots, \ell), \qquad V_{\ell+1} \textcircled{0} K.$$

Since G is $2K_2$-free and K is stable we conclude $V_i \textcircled{1} V_j$ for all i, $j \in \{1,\dots,\ell\}$, $i \neq j$. Now, we distinguish two subcases:

Case 1a. $\bigcup_{i=1}^{\ell} V_i \subseteq L$.
Let $(v_1,\dots,v_\ell) \in V_1 \times \dots \times V_\ell$ and set $D := \{d, v_1,\dots,v_\ell\}$. Clearly, D is a dominating set in G. Suppose, that there is a dominating set D' in G with $|D'| = \ell$. Since K is stable and $|N(x) \cap K| = 1$ for all $x \in N(K)$ by (P3) we need at least ℓ vertices to dominate the vertices in K, i.e.

$$|(K \cup N(K)) \cap D'| \geq |K| = \ell.$$

But now $(K \cup N(K)) \textcircled{0} \{d\}$ leads to a contradiction implying $\gamma(G) = \ell + 1$.

Case 1b. There are $i, j \in \{1,\dots,\ell\}$, $i \neq j$ such that $V_i \cap L \neq \emptyset$ and $V_j \cap (M \cup R) \neq \emptyset$.
Let $(v_1,\dots,v_\ell) \in V_1 \times \dots \times V_\ell$ with $v_i \in L$ and $v_j \in M \cup R$ and set $D := \{v_1,\dots,v_\ell\}$. By construction D will dominate all vertices in

$$\{c,d\} \cup K \cup \bigcup_{i=1}^{\ell} V_i = V \setminus V_{\ell+1}.$$

Let $x \in V_{\ell+1}$. If $x \in L$ then $xv_i \in E$ since L is complete by (P2). If $x \notin L$ then $x \in M \cup R$ and $xv_i \in L$ since otherwise $\{x, d, v_i, k_i\}$ will induce a $2K_2$. This proves that D is a dominating set in G. Analogously to Case 1a one can show that D is minimal, i.e. $\gamma(G) = \ell$.

Case 1c. $L \cap \bigcup_{i=1}^{\ell} V_i = \emptyset$.
This case cannot occur since $b \in L$ has the neighbor a in K.

Case 2. $|K| = 1$.
In this case $D := \{b, d\}$ is a dominating set in G implying $\gamma(G) \leq 2$.

This computation of a minimum dominating set can clearly be implemented in linear time. □

Note, according to our proof of Theorem 3.2.11 one can easily develop a robust algorithm to solve MINIMUM DOMINATING SET on (claw,$2K_2$)-free graphs.

For investigating negative results on the complexity of domination problems on subclasses of claw-free graphs it is useful to consider line graphs — a proper subclass of claw-free graphs. More information (in particular a characterization by forbidden induced subgraphs) one can find in the survey [23].

For a graph $G = (V, E)$ the *line graph* $L(G)$ has as vertices the edge set E of G and two vertices e_1, $e_2 \in E$ of $L(G)$ are adjacent if they have a common endpoint, i.e. $e_1 \cap e_2 \neq \emptyset$. Clearly, every line graph is claw-free because if (e_1, e_2, e_3) is a P_3 in $L(G)$ then e_1 and e_3 must contain different endpoints of e_2.

We call a set $E' \subseteq E$ an *edge dominating set* in G iff E' is a dominating set in $L(G)$, i.e. for all $e \in E$ either $e \in E'$ or $e \cap f \neq \emptyset$ for some $f \in E'$. Let MINIMUM EDGE DOMINATING SET be the problem of computing an edge dominating set of minimum cardinality.

graph class	time complexity	where
bipartite permutation	$O(mn + n^2)$	[118]
bipartite graphs with maximum degree 3	$\mathbb{NP}$–c	[128]
block graphs	LIN	[71]
claw-free chordal graphs	$O(n^2)$	[72]
co-chordal	$O(mn^{1.5} + n^2)$	[118]
locally connected claw-free graphs	$O(n^3)$	[72]
perfect claw-free graphs	$\mathbb{NP}$–c	[72]
planar bipartite graphs	$\mathbb{NP}$–c	[72]
planar cubic graphs	$\mathbb{NP}$–c	[72]
planar graphs with maximum degree 3	$\mathbb{NP}$–c	[128]
trees	LIN	[104, 128]
k-trees	$\mathbb{P}$	[37]

Table 3.2: Complexity of the Edge Domination Problem on special graph classes. Abbreviations: $\mathbb{P}$ for polynomial time, LIN for linear time and $\mathbb{NP}$–c for $\mathbb{NP}$–complete.

This problem has interesting applications in telephone switching networks (see [128]) and among other things the complexity is known for the special graph classes listed in Table 3.2. We show

Theorem 3.2.12 MINIMUM DOMINATING SET *is NP-complete on (claw,diamond,*C_4, K_4*)-free graphs.*

Proof. This result is a consequence of the $\mathbb{NP}$-completeness of MINIMUM EDGE DOMINATING SET on C_4-free bipartite graphs with maximum degree three (see [128][3]). For this we need to investigate what $L(G)$ is H-free, $H \in \{C_4, K_4, \text{diamond}\}$, means in G.

Observation 3.2.13 *The following properties are true:*

1. $\triangle(G) \leq 3$ *if and only if* $L(G)$ *is* K_4*-free.*
2. G *is* $(\textit{paw}, \textit{diamond}, K_4)$*-free if and only if* $L(G)$ *is diamond-free.*
3. G *is* $(C_4, \textit{diamond}, K_4)$*-free if and only if* $L(G)$ *is* C_4*-free.*

Proof.

1. Let (e_1, e_2, e_3, e_4) be a K_4 in $L(G)$. Then (e_1, e_2, e_3) cannot induce a triangle in G since e_4 is adjacent to all elements in $\{e_1, e_2, e_3\}$. Therefore, e_1, e_2, e_3 contain a common

[3] In [128] it is only written that MINIMUM EDGE DOMINATING SET is $\mathbb{NP}$-complete on bipartite graphs with maximum degree three but the graph used in the reduction proof is C_4-free, too.

G_i / G_j	K_3	$2K_2$	claw	co-claw	C_4	paw	diamond	K_4
K_3	$\mathbb{NP}$–c	LIN	LIN	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c
$2K_2$	LIN	$\mathbb{NP}$–c	LIN	$O(n^3)$	$\mathbb{NP}$–c	LIN	LIN	$O(n^3)$
claw	LIN	LIN	$\mathbb{NP}$–c	LIN	$\mathbb{NP}$–c	LIN	$\mathbb{NP}$–c	$\mathbb{NP}$–c
co-claw	$\mathbb{NP}$–c	$O(n^3)$	LIN	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c
C_4	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c
paw	$\mathbb{NP}$–c	LIN	LIN	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c
diamond	$\mathbb{NP}$–c	LIN	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c
K_4	$\mathbb{NP}$–c	$O(n^3)$	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c	$\mathbb{NP}$–c

Figure 3.8: MINIMUM DOMINATING SET on (G_i, G_j)-free graphs. Abbreviations: LIN for linear time and $\mathbb{NP}$–c for $\mathbb{NP}$–complete.

vertex $x \in e_1 \cap e_2 \cap e_3$. Since e_4 is adjacent to e_1, e_2, e_3, vertex x also is an endpoint of e_4 implying $\deg_G(x) \geq 4$. The other direction is clear.

2. Let (e_1, e_2, e_3, e_4) be a diamond in $L(G)$ with nonedge between e_2 and e_4. Since e_1, e_2, e_3 is a clique in $L(G)$ these edges are either a claw (note: not necessary an induced claw!) or a triangle in G. In both cases the remaining edge e_4 will give a paw. Thus, G contains a paw, diamond or K_4.

3. Let (e_1, e_2, e_3, e_4) be a C_4 in $L(G)$. Then, (e_1, e_2, e_3, e_4) must also form a 4-cycle in G. A 4-cycle induces in G a C_4, diamond or K_4. □

By this observation line graphs of C_4-free bipartite graphs with maximum degree three are (claw,diamond,C_4, K_4)-free which settles the proof of Theorem 3.2.12. □

3.2.3 Overview

In this subsection we want to summarize the obtained results.

For one forbidden subgraph Theorem 3.2.1 gives a complete answer to the complexity status of the problem MINIMUM DOMINATING SET.

For two forbidden subgraphs with at most four vertices the complexity status is either polynomial or the status is given in Figure 3.8.

Altogether, we are able now to give the complexity status for all $\mathcal{F}$-free graphs where $\mathcal{F}$ is a set of graphs containing at most four vertices.

Corollary 3.2.14 *Let $\mathcal{F}$ be a set of graphs containing at most four vertices. Then, either* MINIMUM DOMINATING SET *can be solved in polynomial time on $\mathcal{F}$-free graphs or $\mathcal{F}$ is a superclass of one of the following graph classes and* MINIMUM DOMINATING SET *is $\mathbb{NP}$-complete:*

- $(C_4, 2K_2)$-*free graphs,*
- *(claw, C_4, diamond, K_4)-free graphs,*
- *(K_3, C_4)-free graphs.*

3.3 The class $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$— contraction of homogeneous sets

In this section we consider the relation of homogeneous sets to the minimum r–dominating set problem for graphs without an underlying tree structure. We show that one can reduce a homogeneous set to one vertex or to a so–called meta–vertex consisting of two nonadjacent vertices with r–value one. This reduction together with modular decomposition leads to an $O(|V||E|)$ time algorithm for computing a minimum r–dominating set for graphs which can be generated from the one–vertex graph by a finite number of homogeneous extensions (substitution of an arbitrary graph into a vertex) and by attaching pendant vertices (leaves). For distance–hereditary graphs — a proper subclass of this graph class — we even get a linear time algorithm.

In [92] the authors reduce the total dominating set problem to the dominating set problem. They show that for any graph class which is closed under adding false twins an $O(f(n,m))$ algorithm for the dominating set problem leads to an $O(f(n,m))$ algorithm for the total dominating set problem (see Theorem 2.1.2). Therefore we get a linear time algorithm for the total dominating set problem on distance–hereditary graphs.

As mentioned in Section 3.1 the existence of a linear time algorithm for the minimum dominating set problem on distance–hereditary graphs can be proved using another approach: In [68] the authors consider the clique width of some perfect graph classes. They show that for distance–hereditary graphs the clique width is bounded by three and that a 3–expression can be computed in linear time. As consequence (see Theorem 3.1.3 and 3.1.5) MINIMUM DOMINATING SET can be solved on this class in linear time.

This section is organized as follows. After presenting basic definitions and notations we will outline the idea of the algorithm presented in [108] for homogeneous extensions of trees. We hope this gives a better understanding of the more general reduction presented in subsection three. After that this theorem is used to develop efficient algorithms for solving the minimum r–dominating set problem on $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ and distance–hereditary graphs. Finally, some open problems for further research are listed.

3.3.1 Homogeneous Extensions of Graphs

Recall, a set $H \subseteq V$ is called *homogeneous* iff any pair of vertices of H has the same neighborhood outside H:

$$N(u) \cap (V \setminus H) = N(v) \cap (V \setminus H) \qquad \text{for all } u, v \in H.$$

A homogeneous set H is *proper* (or *nontrivial*) iff $0 < |H| < |V|$. With $\mathcal{H}(G)$ we denote the set of all maximal proper homogeneous sets of G.

Let H be a proper homogeneous set of G containing at least two vertices and let $v_H \in H$. Then the graph $\mathrm{HRed}(G, H, v_H)$ obtained from G by deleting $H \setminus \{v_H\}$, i.e. contracting H to a representing vertex v_H, will be called the *homogeneous reduction* of G (via H). The following is easy to prove.

Observation 3.3.1 *Let $G = (V, E)$ be a graph, H a proper homogeneous set in G and $v_H \in H$. Then the distances in G and $G' = \mathrm{HRed}(G, H, v_H)$ fulfill the following properties:*

(1) $d_G(x, y) = d_{G'}(x, y)$ *for all $x, y \in V \setminus H$,*

(2) $d_G(x, H) = d_{G'}(x, v_H)$ *for all $x \in V \setminus H$,*

(3) $d_G(h_1, h_2) \leq 2$ *for all $h_1, h_2 \in H$ if G is connected.*

Conversely, for a connected graph G of at least two vertices the *homogeneous extension* $\mathrm{HExt}(G, v, H)$ of G via a graph H in v is the graph obtained by substituting v by H such that the vertices of H have the same neighbors outside H as v had in G. Note that it is necessary for structural reasons to have at least two vertices in G since otherwise each graph can be represented as the homogeneous extension of a single vertex by itself.

Recall, a graph $G = (V, E)$ is called *prime graph* iff $\emptyset$, V and the singletons of G are the only homogeneous sets in G. For a connected graph G we define $\mathrm{Prim}(G)$ as:

- G if G has at most two vertices.
- If $\mathcal{H}(G)$ is a partition of V then $\mathrm{Prim}(G)$ is the graph obtained by contracting each proper homogeneous set of $\mathcal{H}(G)$ to a representing vertex.
- Otherwise we define $\mathrm{Prim}(G) = K_2$.

Two homogeneous sets H_1, H_2 *overlap* iff their intersection and their mutual differences are nonempty. A homogeneous set H is *overlap–free* iff there is no other homogeneous set overlapping H. Since for any overlap–free homogeneous set $H \subset V$ there is exactly one minimal overlap–free homogeneous set H' containing H properly we obtain a parent function by $\mathrm{parent}(H) := H'$. Thus, using V as root, this gives a tree of homogeneous sets which is called the *module tree* $T_M(G)$. This tree can be computed in linear sequential time (cf. [106], [34]).

In [108] we consider the relationship between the maximal proper and maximal overlap–free homogeneous sets.

Lemma 3.3.2 ([108]) *Let G be a graph with at least two vertices.*

- *If $\mathcal{H}(G)$ is a partition of V then $\mathcal{H}(G)$ is exactly the set of maximal proper overlap–free homogeneous sets of G.*

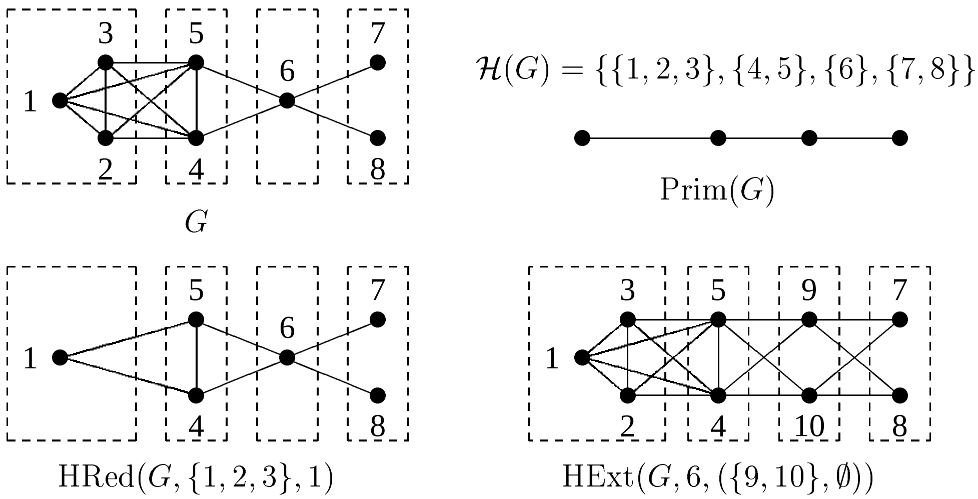


Figure 3.9: Example for the operations HRed, HExt and Prim.

- *If $\mathcal{H}(G)$ is not a partition of V then the maximal proper overlap–free homogeneous sets of G are exactly the connected components $V_1,\ldots,V_\ell$ of the complement $\overline{G}$, and $\ell \geq 3$.*

Thus, using modular decomposition, we can compute in linear time homogeneous sets $H_1,\ldots,H_\ell$ such that $\{H_1,\ldots,H_\ell\}$ is a partition of V and such that reducing each H_i to a representing vertex leads to $\mathrm{Prim}(G)$ (if $\mathcal{H}(G)$ is a partition of $V(G)$ then we take $\mathcal{H}(G)$, otherwise $V_1, \bigcup_{i=2}^{\ell} V_i$ is a partition of $V(G)$ and $\mathrm{Prim}(G) = K_2$).

Next we want to describe graph classes which can be generated from some basic graphs by a finite number of certain operations. Let $\mathcal{B}$ be a set of graphs and $\mathcal{O}$ be a set of operations defined on graphs. Then $\Gamma_{\mathcal{O}}(\mathcal{B})$ is defined as:

(1) Every graph of $\mathcal{B}$ belongs to $\Gamma_{\mathcal{O}}(\mathcal{B})$.

(2) If op is an k–ary operation from $\mathcal{O}$ and $G_1,\ldots,G_k$ are graphs from $\Gamma_{\mathcal{O}}(\mathcal{B})$ then $op(G_1,\ldots,G_k)$ belongs to $\Gamma_{\mathcal{O}}(\mathcal{B})$, too.

Many known graph classes can be described in this way. Indeed, any graph class which can be characterized via dismantling schemes can be represented by the above Γ–notation. Thus, trees, chordal graphs, strongly chordal graphs, dually chordal graphs, distance–hereditary graphs, HHD–free graphs can be described by the Γ–notation. For instance, distance–hereditary graphs are exactly the graphs of $\Gamma_{\{\mathrm{PV,FT,TT}\}}(K_1)$ (cf. [24, 77]). Since any pair of twins is a homogeneous set distance–hereditary graphs are a (proper) subclass of $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$.

In Figure 3.11 the inclusion hierarchy of some graph classes is shown. For definitions and characterizations of these graphs we refer to [23] for an overview, to [12] for dually chordal graphs, to [13] for homogeneously orderable graphs and to [54] for homogeneous graphs. Here we only want to define the MN–operation. A vertex $u \in D(v,1)$ is a *maximum*

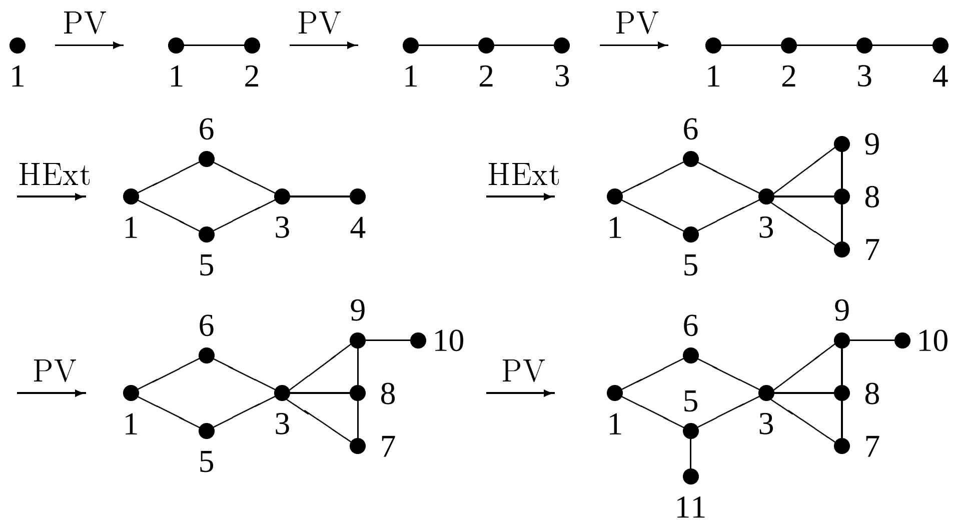


Figure 3.10: Generation of a graph in $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$ which is not in $\Gamma_{\{\mathrm{HExt}\}}$(trees).

neighbor of a vertex v iff $D(u,1) = D(v,2)$, i.e. $D(w,1) \subseteq D(u,1)$ for all $w \in D(v,1)$. If v has a maximum neighbor then it is called *extremal*. Now by MN we denote the operation of adding an extremal vertex.

By the recursive definition of $\Gamma_{\mathcal{O}}(\mathcal{B})$ there is a corresponding generation sequence to each graph G of $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$: We call $((A_1, v_1), \ldots, (A_\ell, v_\ell))$ a *generation sequence* of G iff

(1) G_0 is the one vertex graph with (initial) vertex v_1,

(2) $G_i := \mathrm{HExt}(G_{i-1}, v_i, A_i)$ if $2 \leq |A_i|$, or G_i is obtained by attaching x as pendant vertex to v_i where $A_i = \{x\}$, $i \in \{1, \ldots, \ell\}$,

(3) $G_\ell = G$.

Note that by the definition of homogeneous extensions G_1 is always obtained by attaching a pendant vertex to the vertex of G_0.
The reverse order of a generation sequence is called *reduction sequence*.

Observation 3.3.3 ([36]) *For homogeneous sets H_1 and H_2 of a graph G the following properties are fulfilled:*

1. *$H_1 \cap H_2$ is homogeneous.*
2. *If $H_2 \not\subseteq H_1$ then $H_1 \setminus H_2$ is homogeneous.*
3. *If $H_1 \cap H_2 \neq \emptyset$ then $H_1 \cup H_2$ is homogeneous.*

Lemma 3.3.4 ([109]) *Let G be a graph in $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$.*

(1) *If v is a pendant vertex in G then $G - v \in \Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$.*

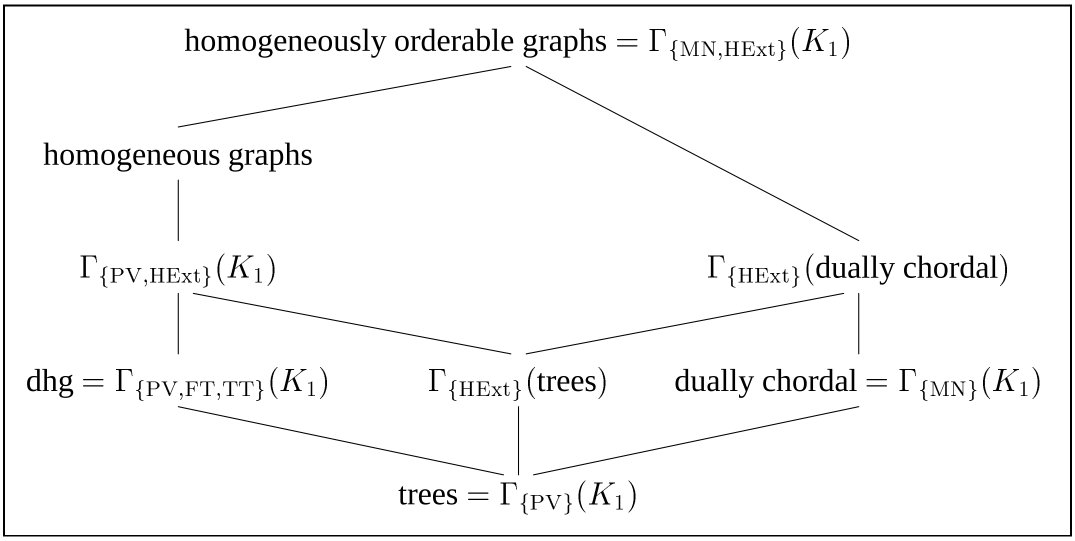


Figure 3.11: Inclusion hierarchy of some graph classes.

(2) *If H is a proper homogeneous set in G and $v_H \in H$ then* $\mathrm{HRed}(G, H, v_H) \in \Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$.

Proof. Let $\sigma = ((A_1, v_1), \dots, (A_\ell, v_\ell))$ be a generation sequence of G.

(1) First assume that v is the initial vertex, i.e. $G_0 = (\{v\}, \emptyset)$. Recall that G_1 is obtained by attaching x as pendant vertex to v, $A_1 = \{x\}$. Since v is a pendant vertex in G no further operations are performed on v. Thus, by starting with x as initial vertex, we can rearrange σ to

$$\sigma' := ((A_2, v_2), \dots, (A_\ell, v_\ell), (\{v\}, x)).$$

Now let i be the position in σ such that $v \in V(G_i)$ but $v \notin V(G_{i-1})$.

If v is attached as pendant vertex in step i, i.e. $A_i = \{v\}$ then

$$\sigma' = ((A_1, v_1), \dots, (A_{i-1}, v_{i-1}), (A_{i+1}, v_{i+1}), \dots, (A_\ell, v_\ell), (A_i, v_i))$$

is a generation sequence of G, too. Thus, $G - v$ is in $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$.

If $v \in A_i$ and $|A_i| \geq 2$ then v is an isolated vertex in $G(A_i)$ and $N_{G_i}(A_i) = \{x\}$. Thus we can generate G by

$$\sigma' = ((A_1, v_1), \dots, (A_{i-1}, v_{i-1}), (A_i \setminus \{v\}, v_i), (\{v\}, x), (A_{i+1}, v_{i+1}), \dots, (A_\ell, v_\ell)),$$

i.e. v can be attached as pendant vertex. The assertion follows by the preceding arguments.

(2) Assume for the contrary that G is a vertex minimal graph in $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ such that there exists a proper homogeneous set H in G with $\mathrm{HRed}(G, H, v_H) \notin \Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$. Then clearly $|H| \geq 2$ and $H \neq A_\ell$.

At first we consider the case $|A_\ell| \geq 2$. Supposing $A_\ell \cap H = \emptyset$ we get a contradiction to the minimality of G by

$$\mathrm{HRed}(G, H, v_H) = \mathrm{HExt}(\mathrm{HRed}(\mathrm{HRed}(G, A_\ell, v_\ell), H, v_H), v_\ell, G(A_\ell)).$$

Therefore $H \cap A_\ell \neq \emptyset$.

If $H \subseteq A_\ell$ we have nothing to show since

$$\mathrm{HRed}(G, H, v_H) = \mathrm{HExt}(\mathrm{HRed}(G, A_\ell, v_\ell), v_\ell, \mathrm{HRed}(G(A_\ell), H, v_H)).$$

So let $H_1 := H \setminus A_\ell \neq \emptyset$. Analogously we may assume $H_2 := A_\ell \setminus H \neq \emptyset$ since otherwise we get

$$\mathrm{HRed}(G, H, v_H) = \mathrm{HRed}(\mathrm{HRed}(G, A_\ell, v_\ell), H_1 \cup \{v_\ell\}, v_H).$$

Note that by Observation 3.3.3 each of the sets H_1, H_2 and $H \cup A_\ell$ is homogeneous. But now

$$\mathrm{HRed}(G, H, v_H) = \mathrm{HExt}(\mathrm{HRed}(\mathrm{HRed}(G, A_\ell, v_\ell), H_1, v_H), v_\ell, G(H_2))$$

gives a contradiction.

In the case $|A_\ell| = 1$ we conclude $v_\ell \in H$ by (1) since G is a vertex minimal counterexample. But then

$$\mathrm{HRed}(G, H, v_H) = \mathrm{HRed}(G - v_\ell, H \setminus \{v_\ell\}, v_H),$$

which settles the proof. □

A direct consequence of the last lemma is

Observation 3.3.5 ([109]) *Let $G \neq K_1$ be a graph in $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$. Then $\mathrm{Prim}(G)$ has at least one pendant vertex.*

Furthermore, we obtain a simple recognition algorithm.

Corollary 3.3.6 ([109]) *Graphs of $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$ can be recognized in $O(|V||E|)$ time.*

Proof. The following procedure recognizes graphs of $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$:

(1) Compute the module tree $T = T_M(G)$ and let w be its root.

(2) If $N_T(w) = \mathcal{H}(G)$ then

- Homogeneously reduce each H of $\mathcal{H}$ to a representing vertex v_H establishing $\mathrm{Prim}(G)$.
- If $\mathrm{Prim}(G)$ has no pendant vertex then stop — G is not in $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$.

- Otherwise, as long as possible dismantle all pendant vertices. Let G' be the obtained graph.
- If $G' = K_1$ then stop — G is in $\Gamma_{\{\text{PV,HExt}\}}(K_1)$. Otherwise go to step (1) with $G = G'$.

(3) Otherwise ($\mathcal{H}(G)$ is not a partition of G) stop — G is in $\Gamma_{\{\text{PV,HExt}\}}(K_1)$.

Since the computation of the module tree takes linear time the whole algorithm runs in $O(|V||E|)$ time. □

As already mentioned in the introduction the minimum dominating set problem on distance–hereditary graphs can be solved in linear time since distance–hereditary graphs have bounded clique width (see [68]). For graphs in $\Gamma_{\{\text{PV,HExt}\}}(K_1)$ this approach cannot be used:

Observation 3.3.7 ([109]) *The class $\Gamma_{\{\text{PV,HExt}\}}(K_1)$ is not of bounded clique width.*

Proof. Let $G = (V, E)$ be a graph of clique width $k \geq 2$ and $x \notin V$. With $G \textcircled{1} x$ we denote the graph $(V \cup \{x\}, E \cup \{xv : v \in V\})$ (see Chapter 2). Then it is easy to see that $G \textcircled{1} x \in \Gamma_{\{\text{PV,HExt}\}}(K_1)$ and that $G \textcircled{1} x$ has clique width k, too. □

3.3.2 The r–domination set problem on homogeneous extensions of trees

In [108] we have seen that homogeneous sets behave very well with respect to domination problems. For a better understanding of the reduction presented in the next subsection we will at first outline the idea of the algorithm presented in [108] for homogeneous extensions of trees.

Let $G = (V, E)$ be a graph with radius function $r : V \to \mathbb{N}$. For every homogeneous set H of G we denote by H^0 the set of vertices of H with r–value zero. By definition H^0 must be included in any r–dominating set of G. Thus, if $H^0 \neq \emptyset$ we may reduce H^0 to a single vertex (which is adjacent to all neighbors of H^0), and at the end of the algorithm we replace this single vertex by the whole set H^0. So we may assume $|H^0| \leq 1$. But now each proper homogeneous set H of G can be r–dominated by at most two vertices (recall $|V(G)| \geq 2$). Hence we transform G into a graph G' consisting of vertices of the following two types:

(1) If there is a vertex v of H which r–dominates H in G then H is r–dominated by a single vertex. In this case v_H is a usual vertex. We define $r(v_H) := \min\{r(v) : v \in H\}$.

(2) In all other cases v_H is a meta–vertex consisting of two nonadjacent inner vertices v_H^1, v_H^2 such that:
If $H^0 \neq \emptyset$ then $r(v_H^1) := 0$ and $r(v_H^2) := 1$, otherwise $r(v_H^1) := r(v_H^2) := 1$.

An example of this transformation is given in Figure 3.12. It is easy to see that an optimal solution for the r–dominating set problem on G can be constructed from an optimal solution for the r–dominating set problem on G'.

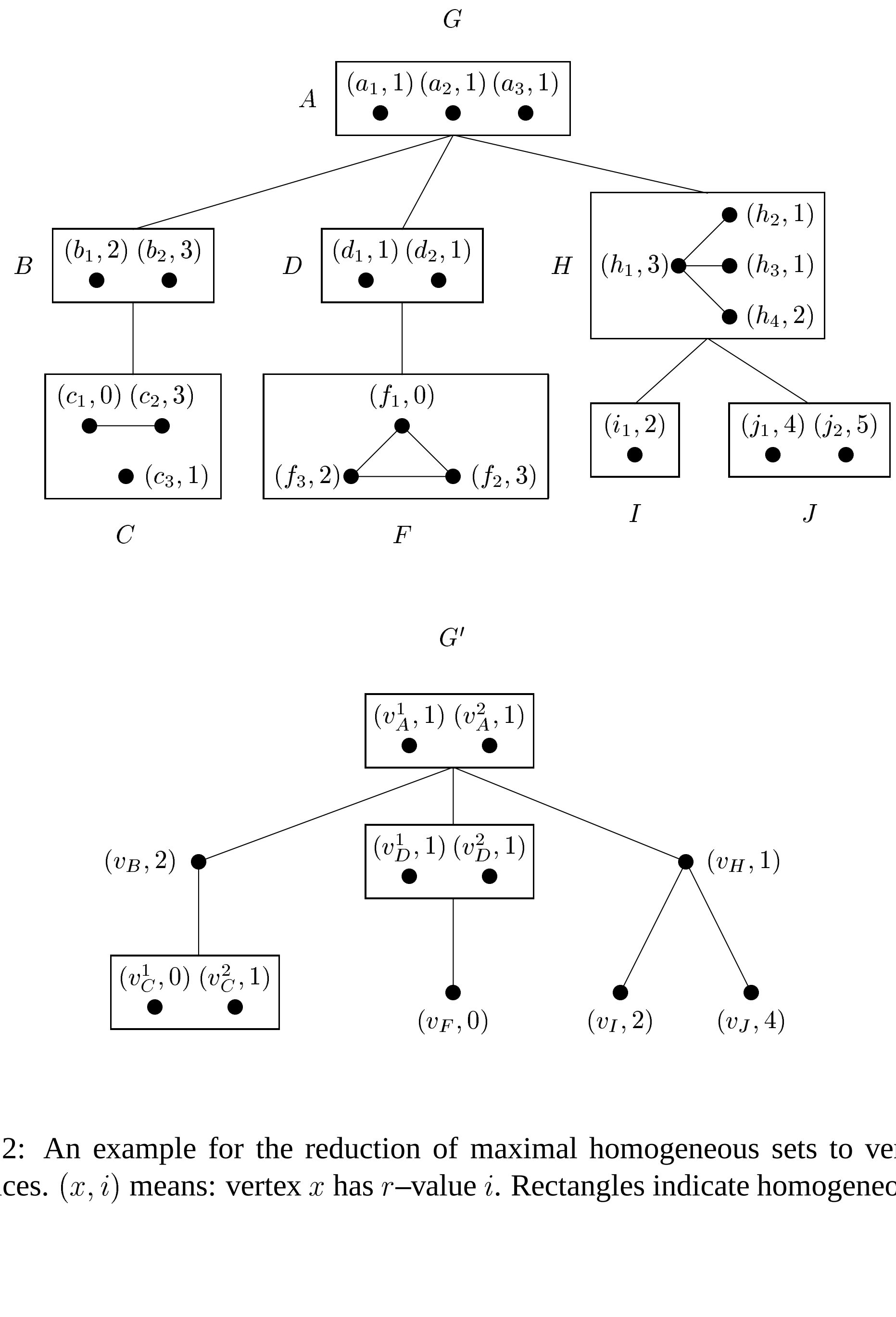


Figure 3.12: An example for the reduction of maximal homogeneous sets to vertices or meta–vertices. (x, i) means: vertex x has r–value i. Rectangles indicate homogeneous sets.

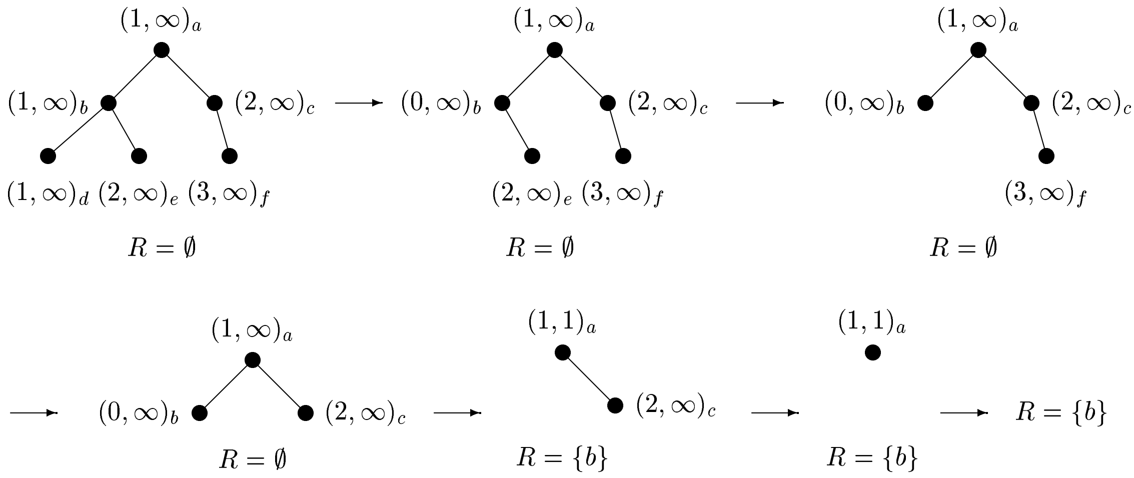


Figure 3.13: An example to the algorithm presented in [9]. $(i, j)_v$ means: vertex v has r–value i and c–value j.

So, for homogeneous extensions of trees, we obtained a tree T of vertices and meta–vertices for which the r–dominating set problem can be solved in linear time. The idea of the algorithm is similar to the one of [116] and [9] for computing r–dominating sets in trees (the algorithm given in [9] works for dually chordal graphs, a proper superclass of trees). In each step the algorithm takes an arbitrary leaf x of the current tree T (i.e. the tree consisting of all unprocessed vertices) and decides if x must be added to the current dominating set R. The basic rule is to choose vertices for domination closest to the root, i.e. as long as possible we remove leaves without adding these ones to the current dominating set R. After that, x is deleted and certain parameters of the father of x will be updated. If the actually graph is empty R is a minimum r–dominating set for the given graph.

As parameters for the vertices we use two values: $r(x)$, $c(x)$. Initially $r(x)$ is the given r–value and $c(x)$ will be initialized with ∞. In each step for the parameters of each leaf x of the current tree T the following properties hold:

- $r(x)$ represents the distance within x and all still undominated vertices of T_x must be dominated in T,
- $c(x)$ indicates the minimum distance of x to a member of the current r–dominating set R in T_x.

So if $c(x) \leq r(x)$ then x is already r–dominated by R. If $c(x) > r(x)$ then x is r–dominated by some vertex z in the current tree if its father y is r'–dominated by z where $r'(y) := \min\{r(y), r(x) - 1\}$. This works well for usual vertices as proved in [9]. An example of this algorithm is given in Figure 3.13.

For meta–vertices we cannot use the above technique. Indeed, it is impossible to decide during the removal of a leaf with a meta–vertex as father whether this leaf has to be added to the r–dominating set or not. To illustrate this problem consider the example in Figure 3.14.

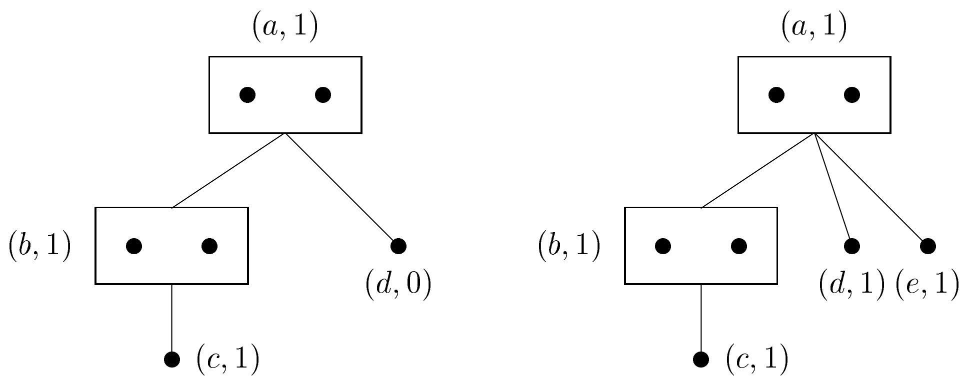


Figure 3.14: The problem with meta–vertices.

In the left graph $\{c,d\}$ is the unique minimum r–dominating set. On the other hand, in the right one there is no minimum r–dominating set containing c.

To avoid this decision we introduce four parameters A_v, B_v, C_v, D_v for meta–vertices v describing all possibilities for local r–domination. Hereby, A_v and B_v contain vertices such that T_v is completely r–dominated by $R \cup A_v$ resp. $R \cup B_v$. The difference between these two sets is that A_v must contain at least one (inner) vertex of v whereas B_v must not contain any vertex of v. On the other hand, the sets C_v and D_v contain vertices such that $T_v \setminus \{v\}$ is r–dominated but not v itself. Note that these sets contain only already processed vertices, i.e. vertices from T_v.

Here we will use the same technique to obtain a reduction theorem for homogeneous sets which is the base for the polynomial time algorithms of the r–dominating set problem on $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ and distance–hereditary graphs presented in the following subsections. Since we have two types of vertices it is convenient to define a more general r–dominating set problem considered in the next subsection.

3.3.3 The generalized r–domination set problem and homogeneous sets

As mentioned in the previous subsection we define a more general r–dominating set problem which additionally reflects the recursion. We are given:

- Finite disjoint sets S — the set of all vertices — and M, a set of markers.
- The set of current vertices $V = V_1 \cup (V_2 \times \{1,2\}) \subseteq S$. For each $v \in V_2$ we write v_1 for $(v,1)$, v_2 for $(v,2)$ and v for $\{v_1, v_2\}$.
- The current graph $G = (V,E)$ such that for all $v \in V_2$ the vertices v_1, v_2 are false twins in G, i.e. $N(v_1) = N(v_2)$. We call the elements of V_2 *meta–vertices*.
- A radius function $r : V_1 \to \mathbb{N} \cup \{\infty\}$.
- An initial set $U_0 \subseteq S \setminus V$.

- For each meta–vertex v we are given sets A_v, B_v, C_v, $D_v \subseteq ((S \cup M) \setminus (V \cup U_0)) \cup v$ which fulfill the following properties:
  - $A_v \cap v \supseteq \{v_1\}$, $C_v \cap v = \{v_1\}$, $(B_v \cup D_v) \cap v = \emptyset$.
  - $|D_v| \geq |A_v| - 2$, $|C_v| \geq |A_v| - 1$, $|B_v| \geq |A_v| - 1$. We call the set X *i–optimal* iff $|X| = |A_v| - i$, $X \in \{B_v, C_v, D_v\}$, $i \in \{1, 2\}$.
  - For distinct meta–vertices x, y the sets $A_x \cup B_x \cup C_x \cup D_x$ and $A_y \cup B_y \cup C_y \cup D_y$ are disjoint.
  - The sets B_v, C_v and D_v need not to be defined, A_v is always defined.

  Further — if D_v is defined — a value $c_D(v) \in \mathbb{N} \cup \{\infty\}$ is given. We call A_v, B_v, C_v, D_v the *set parameters* of v.

A subset U of S is called an *(abstract) dominating set* iff the following properties are fulfilled:

(1) $U_0 \subseteq U$.

(2) (Domination of the meta–vertices)
For each meta–vertex v one of its defined set parameters is completely contained in U. If neither $A_v \subseteq U$ nor $B_v \subseteq U$, then the meta–vertex must be 1–dominated in G, i.e. $N(v) \cap U \neq \emptyset$.

(3) (Domination of the vertices)
For each $v \in V_1$ at least one of the following properties is fulfilled:

(a) v is *r–dominated* in G, i.e. $d_G(v, U \cap V) \leq r(v)$.

(b) There exists a meta–vertex x such that

$$d_G(v,x) + \varepsilon_x \leq r(v), \qquad \varepsilon_x := \begin{cases} 0 & : \; U \cap x \neq \emptyset, \\ 1 & : \; U \cap x = \emptyset, B_x \subseteq U, \\ c_D(x) & : \; U \cap x = \emptyset, B_x \not\subseteq U, D_x \subseteq U. \end{cases}$$

It is easy to verify that with $S = V$ and $M = V_2 = U_0 = \emptyset$ we get the usual r–dominating set problem.

In the following we have given a generalized r–dominating set problem and a (in G) homogeneous set H with $2 \leq |H| < |V|$, such that $x \subseteq H$ for each meta–vertex x with $x \cap H \neq \emptyset$. Let H^0 be the set of vertices of H with r–radius zero. We now describe the reduction of H to a vertex or a meta–vertex.

For each meta–vertex v of V_2 let $\mathcal{S}(v)$ denote the set of defined set parameters associated to v, i.e. $\mathcal{S}(v) \subseteq \{A_v, B_v, C_v, D_v\}$. Note $\mathcal{S}(v) \neq \emptyset$ since A_v is always defined. With $\mathcal{F}(H)$ we denote the set of all set unions of the parameters of all meta–vertices of H, i.e.

$$\mathcal{F}(H) := \begin{cases} \{\emptyset\} & : \; H \cap V_2 = \emptyset, \\ \{\bigcup_{v \in H \cap V_2} S(v) \; : \; S(v) \in \mathcal{S}(v)\} & : \; H \cap V_2 \neq \emptyset. \end{cases}$$

For a set W in $\mathcal{F}(H)$ and a meta–vertex x of H we denote by $W(x)$ the set parameter of x contained in W, i.e. $W(x) \in \mathcal{S}(v)$ with $W(x) \subseteq W$. Let $f := \min\{|W| : W \in \mathcal{F}(H)\}$. A set W of $\mathcal{F}(H)$ with $|W| = f$ has property

$(P1)$ iff W dominates H and $W \cap H \neq \emptyset$,

$(P2)$ iff W dominates H and $W \cap H = \emptyset$,

$(P3)$ iff $W \cap H \neq \emptyset$.

Thus each $(P1)$–set is a $(P3)$–set, too.

At first we consider the case that H does not contain vertices with r–radius zero, i.e. $H^0 = \emptyset$.

Case 1. *$H^0 = \emptyset$ and there is a set W in $\mathcal{F}(H)$ such that $|W| = f$ and W dominates H.*

If there is a $(P1)$–set W in $\mathcal{F}(H)$ then let $x \in W \cap H$. We add $W \setminus \{x\}$ to U_0, reduce H to the vertex x and define $r(x) := 0$.

If there are no $(P1)$–sets but there is a $(P2)$–set W and a $(P3)$–set W' then we reduce H to a meta–vertex h with the following set parameters: $B_h := W$, $C_h := W'$, D_h not defined and $A_h := B_h \cup \{x\}$, where $x \in H$ arbitrary.

Finally, if there are no $(P1)$– and $(P3)$–sets but there is a $(P2)$–set W then we reduce H to a meta–vertex h with the following set parameters: $B_h := W$, C_h, D_h not defined and $A_h := B_h \cup \{x\}$, where $x \in H$ arbitrary.

In Figure 3.15 is given an example for Case 1. The instance G_1 of the r–dominating set problem can be transformed into an abstract dominating set problem G_2 with the following set parameters (the details of the computation of G_2 are given in section 3.3.4)

α	w	y
A_α	$\{w_1, a, b\}$	$\{y_1, e\}$
B_α	$\{a, b\}$	$\{d, e\}$
C_α	n.d.	n.d.
D_α	n.d.	n.d.

Using our reduction relative to the homogeneous set $H = \{w, x, y\}$ we obtain $W := B_w \cup A_y = \{a, b, y_1, e\}$ as an element of $\mathcal{F}(H)$ with minimum cardinality. Since W is a $(P1)$–set we reduce H to the vertex $y_1 \in W \cap H$, add $W \setminus \{y_1\}$ to U_0 and define $r(y_1) := 0$. After that it is not hard to see that the result G_3 leads to the minimum r–dominating set $\{a, b, y_1, e, u\}$ in G_1.

Before considering the remaining cases we show the correctness of Case 1. Note that the remaining correctness proofs can be performed in a similar way and hence are left to the reader.

Lemma 3.3.8 ([109]) *The reductions in Case 1 are correct.*

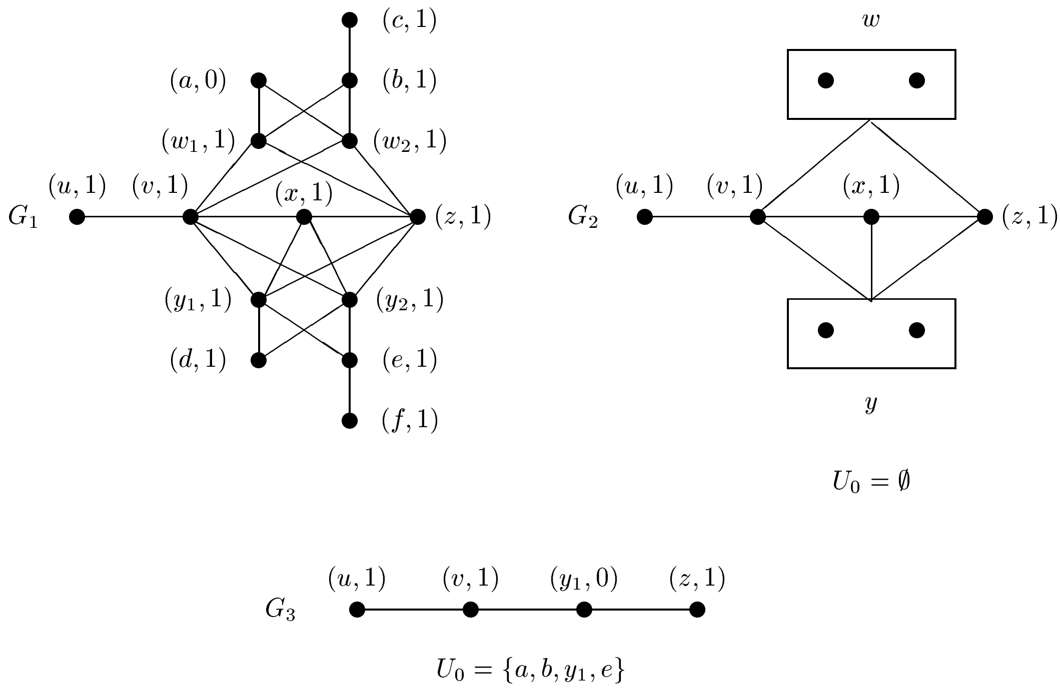


Figure 3.15: An example for Case 1. (α, i) means: vertex α has r–value i.

Proof. At first we have to show that the reduction gives a generalized r–dominating set problem for the reduced graph G'. If there is a $(P1)$–set W then H is reduced to a vertex x and hence there is nothing to show. So consider the case that H is reduced to a meta–vertex h. Obviously, A_h is defined, $|B_h| = |A_h| - 1$ and, provided C_h is defined, $|C_h| = |A_h| - 1$. Since the sets W, W' are set unions of set parameters of H these sets are disjoint from any union of set parameters of a meta–vertex outside H. Further, since all set parameters of meta–vertices contain only vertices outside the current graph (up to the meta–vertex itself) no set parameter of any meta–vertex of $V_2 \setminus H$ contains any vertex of H. Thus in G' the set parameters of different meta–vertices remain disjoint.

Next we have to show that any minimum r–dominating set U of G' is a minimum r–dominating set in G too. First consider the case that there is a $(P1)$–set W. Then H is reduced to a vertex x and $r(x) = 0$ in G'. Furthermore, $W \setminus \{x\}$ is added to U_0. Thus any minimum r–dominating set U in G' must contain W and hence, U is a r–dominating set in G too. Now suppose that U is not minimum in G, i.e. there is a set U' r–dominating G such that $|U'| < |U|$. By definition we get $|U' \cap H| \geq f$. Let $U'' := (U' \setminus H) \cup W$. Obviously U'' is a r–dominating set in G' and hence $|U''| \geq |U|$. But now $|U| \leq |U''| \leq |U'| - f + |W| = |U'|$ gives a contradiction. Thus U is a minimum r–dominating set in G.

Now consider the case that there are no $(P1)$–sets. Here H is reduced to a meta–vertex h. Let U be a r–dominating set of G. By definition, U must include at least one of the set parameters of h, and if neither A_h nor B_h is contained in U then $N(h) \cap U \neq \emptyset$. Since the set parameters of h are defined via W, W' the set U r–dominates G too. If U is minimum in

Algorithm MARK

$(R1)$ If D_x is 2–optimal then we mark D_x.

$(R2)$ Else, if C_x is 1–optimal and x has a neighbor y in H which is a meta–vertex and either

- C_y is 1–optimal and D_y is not 2–optimal, or
- A_y is optimal (i.e. neither D_y is i–optimal, $i = 1, 2$, nor B_y and C_y are 1–optimal),

then we mark C_x.

$(R3)$ Else, if B_x is 1–optimal then we mark B_x.

$(R4)$ Else, if D_x is 1–optimal then we mark D_x.

$(R5)$ Else, if C_x is 1–optimal then we mark C_x.

$(R6)$ Otherwise we mark A_x.

G' then U is minimum in G which can be proved as before. □

Next we consider the time bound of the reduction in Case 1.

Lemma 3.3.9 ([109]) *It can be checked in linear time whether there are $(P1)$–$(P3)$–sets. Furthermore, such sets can be computed within the same time bound.*

Proof. First note that property $(P3)$ can be tested in linear time since we have only to check whether A_x or C_x is optimal for a meta–vertex x of H.

Now we want to check properties $(P1)$ and $(P2)$. For each meta–vertex x we mark a set parameter of x according to algorithm MARK.

Let W be the union of all marked set parameters. Note that $W(x) = C_x$ via rule $(R2)$ immediately implies that x is dominated by W.

By stepping through the neighborhoods of the vertices of H we can easily check in linear time whether W has property $(P1)$ or $(P2)$. So it remains to show the correctness of the algorithm.

Claim 1. *If W' is a set in $\mathcal{F}(H)$ such that $|W'| = f$ then $|W(x)| = |W'(x)|$ for all meta–vertices x of H.*

By the rules of algorithm MARK we immediately conclude $|W(x)| \leq |W'(x)|$ for all meta–vertices x of H. Thus $|W| = |W'|$ settles the proof.

In particular, if D_x is 2–optimal then $W(x) = W'(x) = D_x$, and, if $W'(x) = A_x$ then $W(x) = A_x$.

Claim 2. *If W is not $(P1)$ then there are no $(P1)$–sets in $\mathcal{F}(H)$.*

Assume for the contrary that W' is a $(P1)$–set in $\mathcal{F}(H)$. We will show that then W is a $(P1)$–set too. We may choose W' such that it contains a maximum number of C–sets of meta–vertices in H. Further let x be an arbitrary meta–vertex in H.

By Claim 1 we may assume that D_x is not 2–optimal and $W'(x) \neq A_x$.

If $W'(x) = C_x$ then C_x is 1–optimal by Claim 1. Since W' is a $(P1)$–set there must be a neighbor y of x in $H \cap V_2$ such that $W'(y) \in \{C_y, A_y\}$. Therefore $W(x) = C_x$ by rule $(R2)$.

Now let $W'(x) \in \{B_x, D_x\}$. We will show $W(x) \in \{B_x, D_x\}$ too. Supposing $W(x) = C_x$ rule $(R2)$ of algorithm MARK must be applied. Hence, there is a neighbor y in H such that C_y is 1–optimal and D_y is not 2–optimal, or A_y is optimal. If A_y is optimal then we may replace $W'(x)$ by C_x, if C_y is 1–optimal we can take $W'(x) = C_x$ and $W'(y) = C_y$ yielding a contradiction to the choice of W'.

Therefore $W(x) \in \{B_x, D_x\}$. In particular, if $W'(x) = B_x$ then $W(x) = B_x$.

Altogether it follows that W is a $(P1)$–set — a contradiction.

Claim 3. *If W is neither $(P1)$ nor $(P2)$ then there are no $(P2)$–sets in $\mathcal{F}(H)$.*

First note that by Claim 2 there are no $(P1)$–sets in $\mathcal{F}(H)$. Now assume for the contrary that there is a $(P2)$–set W' in $\mathcal{F}(H)$. Obviously, $W'(x) = B_x$ for all meta–vertices x in H. By Claim 1, for each meta–vertex x in H the set B_x is 1–optimal and the set D_x is not 2–optimal. Further, rule $(R2)$ cannot be applied for any meta–vertex x in H since otherwise we can construct a $(P1)$–set by replacing B_x and B_y by C_x and C_y.

Therefore $W = W'$ — a contradiction.

Claim 2 and 3 settle the proof of Lemma 3.3.9. □

Case 2. *$H^0 = \emptyset$ and there are no $(P1)$– and $(P2)$–sets in $\mathcal{F}(H)$.*

If there is a $(P3)$–set W then we reduce H to a meta–vertex h with the following set parameters: $C_h := W$, B_h, D_h not defined and $A_h := C_h \cup \{m\}$, where $m \in M$ is a marker.

If m is taken to the dominating set of the rest graph, then we substitute m by an arbitrary vertex of $N(H)$.

Now consider the case that there are no (P_1)–, $(P2)$– and no $(P3)$–sets.

Case A. *H contains no meta–vertices.*

Then $\mathcal{F}(H) = \{\emptyset\}$. Since there are no $(P2)$–sets there is at least one vertex $h \in H$ with $r(h) \neq \infty$. Let H' be the set of vertices h of H such that $r(h) = 1$, or the empty set if there are no such vertices. If there is a vertex $h \in H$ such that $H' \subseteq D(h, 1)$ then we reduce H to h and define $r(h) := \min\{r(x) : x \in H \wedge r(x) \leq c(x)\}$.

If $H' \not\subseteq D(h,1)$ for each $h \in H$ then, in particular, H' contains two nonadjacent vertices. We reduce H to a meta–vertex h with the following set parameters: $A_h = h$, B_h not defined, $C_h = \{h_1\}$, $D_h = \emptyset$, $c_D(h) = \infty$. Let U' be a dominating set in the rest graph. Then

$$U := \begin{cases} U' & : \ |U' \cap h| = 0, \\ (U' \setminus h) \cup \{x\} & : \ |U' \cap h| = 1, x \in H, \\ U' \cup \{x, y\} & : \ |U' \cap h| = 2, x \in H, y \in N(H) \end{cases}$$

is a dominating set in G.

Case B. *H contains meta–vertices.*

Then we reduce H to a meta–vertex h. Note that each minimum set–union of the set parameters of the meta–vertices consists only of B– and D–sets. Further note that there is at least one meta–vertex x in H such that $W(x) = D_x$.

Let W^* be the set obtained from W by replacing any set D_x by B_x provided $|D_x| = |B_x|$. Among all sets $W \in \mathcal{F}(H)$ with $|W| = f$ choose a set W such that the union of all B–sets of W^* is maximal. Define $D_h := W^*$ and

$$c_D(h) := \begin{cases} 1 & : \ \exists x \in V_2 \cap H : W^*(x) = B_x, \\ \min\{c_D(x) \ : \ D_x \subseteq W^*\} & : \ \text{otherwise.} \end{cases}$$

If there are sets $W_1 \in \mathcal{F}(H)$ and $W_2 \subseteq V_1 \cap H$ such that $|W_1 \cup W_2| = f + 1$, $W_1 \cup W_2$ dominates H and $(W_1 \cup W_2) \cap H \neq \emptyset$ then let $A_h := W_1 \cup W_2$ and B_h, C_h be undefined. Note $0 \leq |W_2| \leq 1$.

Else, if there is a set $W \in \mathcal{F}(H)$ such that $|W| = f + 1$, W dominates H and $W \cap H = \emptyset$ then let $x \in H$ be an arbitrary vertex. We define $B_h := W$, $C_h := D_h \cup \{x\}$, $A_h := B_h \cup \{x\}$.

Otherwise, let $x \in H$ be an arbitrary vertex. We define $C_h := W^* \cup \{x\}$, $A_h := C_h \cup \{m\}$, B_h is undefined, where $m \in M$ is a marker. If m is taken to the dominating set of the reduced graph, then we substitute m by an arbitrary vertex of $N(H)$ in G.

Lemma 3.3.10 ([109]) *Case 2 can be handled in linear time.*

Proof. First recall, that property $(P3)$ can be tested in linear time. Further, in Case A all decisions can be made by stepping through the neighborhoods of the vertices of H.

So let us consider Case B. It is easy to see that each set $W \in \mathcal{F}(H)$ with $|W| = f$ contains all 2–optimal D–sets of H and a remaining collection of 1–optimal B– and D–sets. Thus, applying algorithm MARK gives the desired set W^*. Now we check whether there is a vertex $x \in H \cap V_1$ such that $W^* \cup \{x\}$ dominates H. If there is such a vertex then define $W_1 := W^*$ and $W_2 := \{x\}$. Otherwise, by considering the meta–vertices v for which $W^*(v)$ is either B_v or an 1–optimal D_v, we check whether we can replace the 1–optimal set $W^*(v)$ by A_v. If there is such a meta–vertex v then define $W_2 := \emptyset$ and $W_1 := (W^* \setminus W^*(v)) \cup A_v$.

All these checks can be easily done in linear time by stepping through the neighborhoods of the vertices of H. So the case that we have sets W_1, W_2 such that $|W_1 \cup W_2| = f + 1$, $W_1 \cup W_2$ dominates H and $(W_1 \cup W_2) \cap H \neq \emptyset$ is handled.

If we do not find such sets W_1, W_2 then we check whether we can replace one 2–optimal set D_v by an 1–optimal set B_v in order to get a dominating set for H. Clearly, this is only possible if there is exactly one 2–optimal D–set in W^*. So we have to compute the number of 2–optimal D–sets (this can be done while running algorithm MARK) and then, if we have exactly one such set D_v, it must be checked whether B_v is 1–optimal and, if so, whether $(W^* \setminus D_v) \cup B_v$ dominates H. Again, this can be done in linear time. □

Case 3. $H^0 \neq \emptyset$.

If there is a set W of $\mathcal{F}(H)$ such that $|W| = f$ and $W \cup H^0$ dominates H then we reduce H to a vertex h_0, $h_0 \in H^0$, and define $r(h_0) = 0$. Furthermore, we add $(W \cup H^0) \setminus \{h_0\}$ to the set U_0.

Otherwise, we reduce H to a meta–vertex h with the following set parameters: $C_h := W \cup H^0$, B_h and D_h are undefined, $A_h := C_h \cup \{m\}$ where $m \in M$ is a marker. If m is taken to the dominating set of the reduced graph, then we substitute m by an arbitrary vertex of $N(H)$ in G.

Summarizing the above results we obtain

Theorem 3.3.11 ([109]) *For solving the generalized domination problem for a graph G we can contract a homogeneous set H to a vertex or a meta–vertex v and extend a dominating set of $\mathrm{HRed}(G, H, v)$ to a dominating set of G in linear time in $G(H)$.*

3.3.4 The r–dominating set problem on $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$

For (not necessarily defined) sets $S_1, \ldots, S_\ell$ let $\min(S_1, \ldots, S_\ell)$ be undefined if each set S_i, $i \in \{1, \ldots, \ell\}$ is not defined, otherwise $\min(S_1, \ldots, S_\ell)$ denote a defined set of $\{S_1, \ldots, S_\ell\}$ with minimum cardinality.

Theorem 3.3.12 ([109]) *The r–dominating set problem on $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ can be solved in time $O(|V||E|)$.*

Proof. Let $G \in \Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ be a graph with at least two vertices. By using modular decomposition we compute proper homogeneous sets $H_1, \ldots, H_\ell$ such that $\{H_1, \ldots, H_\ell\}$ is a partition of V and such that reducing each H_i to a representing vertex leads to $\mathrm{Prim}(G)$. By Theorem 3.3.11 we can reduce each H_i, $i \in \{1, \ldots, \ell\}$, to a vertex or a meta–vertex.

According to Observation 3.3.5 $\mathrm{Prim}(G)$ has a pendant vertex y. Let x be the neighbor of y in $\mathrm{Prim}(G)$. In the following we show that we can delete y. We consider the following four cases :

Case 1. x and y are meta–vertices.

We update the set parameters of x in the following way:

$$A_x := \min(A_x \cup \min(A_y, B_y, C_y, D_y),\; C_x \cup \min(A_y, C_y)),$$

$$B_x := \min(B_x \cup \min(A_y, B_y),\; D_x \cup A_y)$$

and

$$C_x := C_x \cup \min(B_y, D_y), \qquad D_x := D_x \cup B_y, \qquad c_D(x) := \min(c_D(x), 2).$$

Case 2. x is a meta–vertex and y is a vertex.

If $r(y) = \infty$, i.e. y is already dominated, then

$$A_x := \min(A_x, C_x \cup \{y\}), \quad B_x := \min(B_x, D_x \cup \{y\}), \quad C_x := C_x, \quad D_x := D_x,$$

and $c_D(x) := c_D(x)$.

Next, if $r(y) = 0$ then

$$A_x := \{y\} \cup \min(A_x, C_x), \quad B_x := \{y\} \cup \min(B_x, D_x),$$

where C_x, D_x remain undefined.

Finally, for $\infty \neq r(y) \geq 1$ we update the set parameters of x in the following way:

$$A_x := \min(A_x, C_x \cup \{y\}), \quad C_x := C_x,$$

$$B_x := \begin{cases} \min(B_x, D_x \cup \{y\}) & : \quad r(y) \geq 2, \\ \min(B_x \cup \{y\}, D_x \cup \{y\}) & : \quad r(y) = 1 \end{cases}$$

and

$$D_x := \begin{cases} D_x, & : \quad c_D(x) + 1 \leq r(y), \\ \text{n.d.} & : \quad c_D(x) + 1 > r(y). \end{cases}$$

Case 3. y is a meta–vertex and x is a vertex.

After deleting y vertex x will be a meta–vertex in the remaining graph with $A_x := \{x\} \cup \min(A_y, B_y, C_y, D_y)$ and C_x undefined.

If $r(x) = \infty$ then $B_x := \min(A_y, B_y)$ and D_x remains undefined.

For $r(x) = 0$ both sets B_x and D_x are not defined.

Finally, if $\infty \neq r(x) \geq 1$ then we define

$$B_x := \begin{cases} \min(A_y, B_y) & : \quad r(x) \geq 2, \\ A_y & : \quad r(x) = 1, \end{cases} \qquad D_x := \begin{cases} \text{n.d.} & : \quad r(x) \geq 2, \\ B_y & : \quad r(x) = 1. \end{cases}$$

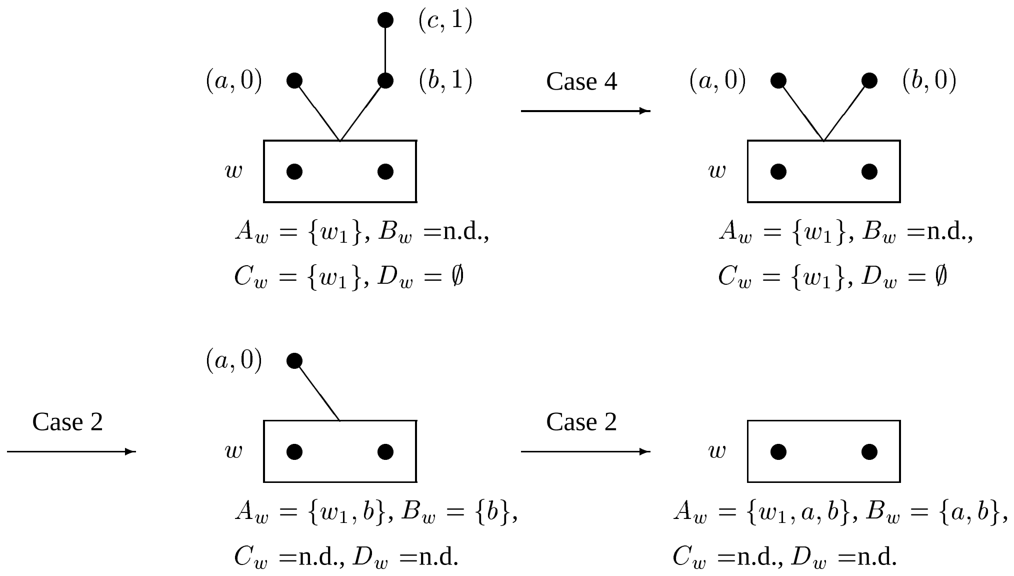


Figure 3.16: An example for the proof of Theorem 3.3.12.

Case 4. x and y are vertices.

If $r(y) = \infty$, i.e. y is already dominated, then we need not to update any parameter of x.

For $r(y) = 0$ we replace x by a meta–vertex with the following set parameters: If $r(x) = 0$ then $A_x := \{x, y\}$, B_x, C_x and D_x are undefined. Otherwise $A_x := \{x, y\}$, $B_x := \{y\}$, C_x and D_x are undefined.

Finally, if $\infty \neq r(y) \geq 1$ then we define

$$r(x) := \begin{cases} r(y) - 1 & : \quad r(x) = \infty, \\ \min(r(x), r(y) - 1) & : \quad r(x) \neq \infty. \end{cases}$$

Note, that by Lemma 3.3.4 $\mathrm{Prim}(G)$ is in $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ too. If $\mathrm{Prim}(G)$ is the one vertex graph with vertex x, then a minimum r–dominating set R of G is constructed as follows :

If x is a meta–vertex, then $R := \min\{A_x, B_x\}$. If x is a vertex then we define $R := U_0$ provided $r(x) = \infty$ and $U_0 \neq \emptyset$; otherwise let $R := U_0 \cup \{x\}$.

If $\mathrm{Prim}(G)$ has at least two vertices then, by Observation 3.3.5, $\mathrm{Prim}(G)$ has at least one pendant vertex. As long as possible we delete pendant vertices as described in the above reduction. Let G' be the remaining graph. By Lemma 3.3.4 G' is in $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ too. If G' is the one vertex graph then we are done as before. Otherwise, we repeat the whole procedure by starting with modular decomposition of G'. So, in worst case $|V|$ times modular decomposition ($O(|V| + |E|)$ time, see [106, 34]) has to be performed.

Altogether this leads to a $O(|V||E|)$ time algorithm for computing a minimum r–dominating set for any graph in $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$. □

Since obviously $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$ is closed under adding false twins by Theorem 2.1.2 we get the following

Corollary 3.3.13 ([109]) *The total dominating set problem on $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$ can be solved in time $O(|V||E|)$.*

3.3.5 The r–dominating set problem on distance–hereditary graphs

Note that distance–hereditary graphs are a proper subclass of $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$. Indeed, the 3–fan (a P_4 with a dominating vertex) is an example for a graph in $\Gamma_{\{\mathrm{PV},\mathrm{HExt}\}}(K_1)$ which is not distance–hereditary.

Recall, that distance–hereditary graphs can be generated from a single vertex by a finite number of one vertex extensions (cf. [24, 77]), i.e. by attaching pendant vertices or twins. Since a pair of twins is a homogeneous set we can use an one vertex extension sequence instead of modular decomposition. Note, that in [77] it is shown that such a sequence can be computed in linear time. Thus we get

Theorem 3.3.14 ([109]) *For a distance–hereditary graph a minimum r–dominating set can be computed in linear time.*

Proof. At first we compute an one vertex extension sequence τ using the linear time algorithm presented in [77].

Now we process τ in reverse order using the update rules of Theorem 3.3.11 for twins and Theorem 3.3.12 for pendant vertices. Note that algorithm MARK runs in constant time since the homogeneous sets are pairs of twins, i.e. sets of constant size two.

In order to get linear running time we store the set parameters of meta–vertices in linked lists. Additionally we store the cardinalities of the set parameters. Hereby undefined sets are considered as sets of infinite cardinality.

Since the update of the set parameters is a simple linking of pointers the algorithm runs in linear time. □

Since distance–hereditary graphs are closed under adding false twins (see [24]) by Theorem 2.1.2 we get the following

Corollary 3.3.15 ([109]) *The total dominating set problem on distance–hereditary graphs can be solved in linear time.*

3.3.6 Concluding Remarks

It would be interesting if the results of section 3.3.3 can be used to solve the r–dominating set problem on homogeneously orderable graphs. These graphs were introduced in [13] as a common generalization of dually chordal and distance–hereditary graphs. For the r–dominating clique and the connected r–dominating set problem on this class polynomial time algorithms are given in [55], the r–dominating set problem is still open.

In homogeneously orderable graphs ($= \Gamma_{\{\mathrm{MN,HExt}\}}(K_1)$) we have to consider extremal vertices instead of pendant vertices. Recall, that a vertex v of a graph G is extremal iff v has a maximum neighbor, i.e. there is a vertex $u \in D(v,1)$ such that $D(u,1) = D(v,2)$.

Analogously to Observation 3.3.5 and Lemma 3.3.4 we have

Observation 3.3.16 ([13]) *Let G be a homogeneously orderable graph. Then $\mathrm{Prim}(G)$ has at least one extremal vertex.*

Furthermore,

Lemma 3.3.17 ([13]) *Let G be a homogeneously orderable graph.*

(1) If H is a proper homogeneous set in G and $v_H \in H$ then the graph $\mathrm{HRed}(G, H, v_H)$ is homogeneously orderable, too.

(2) If v is an extremal vertex in G then $G - v$ is homogeneously orderable, too.

Thus, by point (1) of the preceding Lemma and by Theorem 3.3.11, it remains to consider the removal of extremal vertices.

Appendix A

Complexity of Domination Problems

In this chapter we list the complexity of some domination problems on different graph classes but we declare no claim to completeness.

In the last years at our institute we have developed an information system on graph class inclusions (ISGCI) available via internet under the URL

`http://www.informatik.uni-rostock.de/~gdb/isgci/Isgci.html`

to keep an updated knowledge base of graph classes and their inclusions. The user has the possibility to ask queries about inclusions of classes and to draw inclusion hierarchies for selected classes.

Up to now, the system does not contain any information about the complexity of concrete graph theoretic problems on graph classes contained in the database. It would be very interesting to add known results for well known graph problems such that the user can ask queries. As a starting point the information about MINIMUM DOMINATING SET provided in this chapter can be taken.

Such a system can support research in the following directions:

1. Nearly all published papers are not up–to–date since the referring process takes a long time. An information system is dynamic if the database is updated regularly.

2. The complexity of a problem can be displayed more clearly in an inclusion hierarchy of by the user selected classes (see Figure A.1 for an example).

3. Such a system can be used to find borders on which the complexity of a problem π changes, i.e. classes $\mathcal{C}_1$, $\mathcal{C}_2$ such that $\mathcal{C}_1 \subset \mathcal{C}_2$ and π can be solved efficiently on $\mathcal{C}_1$ but π remains $\mathbb{NP}$–complete on $\mathcal{C}_2$. Then, it is interesting to introduce a new class $\mathcal{C}$ between $\mathcal{C}_1, \mathcal{C}_2$, i.e. $\mathcal{C}_1 \subset \mathcal{C} \subset \mathcal{C}_2$, such that π can be solved efficiently on $\mathcal{C}$, too.

4. A user can search for open problems. Hereby, maybe results for close-by classes can be generalized.

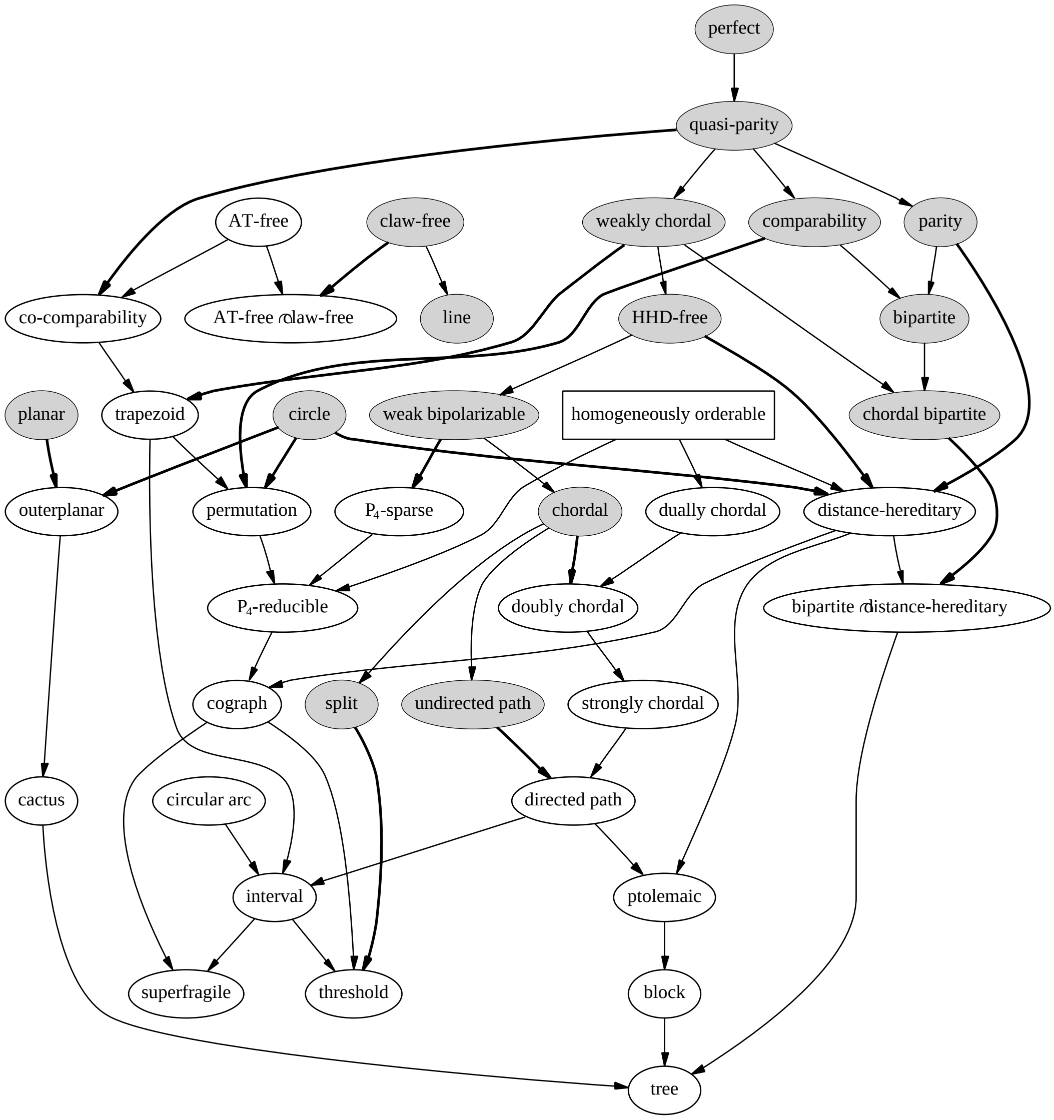


Figure A.1: The complexity of the minimum dominating set problem on some selected graph classes. Hereby, a filled (resp. unfilled) ellipse indicates that the problem is $\mathbb{NP}$-complete (resp. can be solved in polynomial time) and an unfilled box indicates that the complexity status is open. Edges from a class on which the problem remains $\mathbb{NP}$-complete to a class where the problem can be solved in polynomial time are drawn in bold style.

A.1 Abbreviations for problems

Abbreviation	Problem
DS	Minimum cardinality dominating set problem
RDS	Minimum cardinality r–dominating set problem
PDS	Minimum cardinality perfect dominating set problem
DC–ex	Minimum cardinality dominating clique problem (Existence)
DC–comp	Minimum cardinality dominating clique problem (Computation)
RDC–ex	Minimum cardinality r–dominating clique problem (Existence)
RDC–comp	Minimum cardinality r–dominating clique problem (Computation)
IDS	Minimum cardinality independent dominating set problem
IRDS	Minimum cardinality independent r–dominating set problem
IPDS	Minimum cardinality independent perfect dominating set problem
TDS	Minimum cardinality total dominating set problem
TRDS	Minimum cardinality total r–dominating set problem
TPDS	Minimum cardinality total perfect dominating set problem
CDS	Minimum cardinality connected dominating set problem
CRDS	Minimum cardinality connected r–dominating set problem
CPDS	Minimum cardinality connected perfect dominating set problem
WDS	Minimum weighted dominating set problem
WPDS	Minimum weighted perfect dominating set problem
WDC–ex	Minimum weighted dominating clique problem (Existence)
WDC–comp	Minimum weighted dominating clique problem (Computation)
WIDS	Minimum weighted independent dominating set problem
WIPDS	Minimum weighted independent perfect domination problem
WCDS	Minimum weighted connected dominating set problem
WCPDS	Minimum weighted connected perfect dominating set problem
WTDS	Minimum weighted total dominating set problem
WTPDS	Minimum weighted total perfect dominating set problem
STEINER	Minimum cardinality Steiner Tree problem

A.2 Complexity on graph classes

For definitions, characterizations and recognition of the mentioned graph classes we refer to [23]. For time complexity we use the following abbreviations:

$\mathbb{P}$	*for*	polynomial time,
$\mathbb{NP}$–c	*for*	$\mathbb{NP}$–complete.

A.2.1 1–CUBs

Problem	Time complexity	References
DS	$\mathbb{P}$	[47]
DC–ex	$\mathbb{P}$	[47]
DC–comp	$\mathbb{P}$	[47]
TDS	$\mathbb{P}$	[47]
CDS	$\mathbb{P}$	[47]

A.2.2 2–CUBs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[47]
DC–comp	$\mathbb{NP}$–c	[47]
TDS	$\mathbb{NP}$–c	[47]
CDS	$\mathbb{NP}$–c	[47]

A.2.3 AT–free graphs

co–comparability graphs $\subset$ AT–free graphs

Problem	Time complexity	References
DS	$O(n^6)$	[89]
DC–ex	$\mathbb{NP}$–c	see co–comparability graphs
IDS	$O(n^2(\lvert E(\overline{G})\rvert + 1))$	[19]
IPDS	$O(n^2(\lvert E(\overline{G})\rvert + 1))$	[19]
TDS	$O(n^6)$	[89]
CDS	$O(n^3)$	[26]
	$O(n + m)$	if $\mathrm{diam}(G) > 3$, [44]
WDS	$\mathbb{NP}$–c	see co–comparability graphs
WCDS	$\mathbb{NP}$–c	see co–comparability graphs
WTDS	$\mathbb{NP}$–c	see co–comparability graphs
STEINER	$O(n^3)$	[26]

A.2.4 bipartite graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[49], see planar bipartite graphs
PDS	$\mathbb{NP}$–c	[129]
IDS	$\mathbb{NP}$–c	[45], see planar bipartite graphs
IPDS	$\mathbb{NP}$–c	[125, 131]
TDS	$\mathbb{NP}$–c	[111]
TPDS	$\mathbb{NP}$–c	[125, 131]
CDS	$\mathbb{NP}$–c	[111], see planar bipartite graphs
CPDS	$\mathbb{NP}$–c	[125, 131]
WDC–ex	$O(n+m)$	[17], trivial since $\omega(G) \leq 2$
WDC–comp	$O(n+m)$	[17], trivial since $\omega(G) \leq 2$
STEINER	$\mathbb{NP}$–c	see planar bipartite graphs

A.2.5 chordal graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	see split graphs, undirected path graphs
PDS	$\mathbb{NP}$–c	[129]
DC–ex	$O(nm)$	[90]
DC–comp	$\mathbb{NP}$–c	[17]
RDC–ex	$O(nm)$	[51]
IDS	$O(n+m)$	[62]
IPDS	$\mathbb{NP}$–c	[125, 131]
TDS	$\mathbb{NP}$–c	see split graphs
TPDS	$\mathbb{NP}$–c	[125, 131]
CDS	$\mathbb{NP}$–c	see split graphs
WIDS	$\mathbb{NP}$–c	[29]
	$O(n+m)$	[62], for vertex weights in $\{0,1\}$
WCPDS	$O(n+m)$	[38]
STEINER	$\mathbb{NP}$–c	see split graphs

In [90] it is shown that a chordal graph has a dominating clique if and only if $\mathrm{diam}(G) \leq 3$.

A.2.6 chordal bipartite graphs

chordal bipartite graphs $\subset$ bipartite graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[103]
PDS	$\mathbb{NP}$–c	[100]
IDS	$\mathbb{NP}$–c	[53]
TDS	$\mathbb{P}$	[53]
TPDS	$\mathbb{NP}$–c	[120]
CDS	$\mathbb{NP}$–c	[103]
WDC–ex	$O(n+m)$	see bipartite graphs
WDC–comp	$O(n+m)$	see bipartite graphs
STEINER	$\mathbb{NP}$–c	[103]

A.2.7 circle graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[84]
DC–comp	$O(nm)$	[84]
TDS	$\mathbb{NP}$–c	[84]
CDS	$\mathbb{NP}$–c	[84]

A.2.8 circular–arc graphs

In [101] a linear time recognition algorithm for circular–arc graphs is given. A circular–arc model for a circular–arc graph can be computed in the same time bound.

Problem	Time complexity	References
DS	$O(n)$	[80], given a circular–arc model
WDS	$O(n+m)$	[30], given a circular–arc model
WIDS	$O(n+m)$	[30], given a circular–arc model
WCDS	$O(n+m)$	[30], given a circular–arc model
WTDS	$O(n+m)$	[30], given a circular–arc model

A.2.9 claw–free AT–free graphs

Problem	Time complexity	References
DS	$O(n+m)$	[73]
IDS	$O(n+m)$	[73]
IPDS	$O(n^2(\|E(\overline{G})\|+1))$	see AT–free graphs
TDS	$O(n^6)$	see AT–free graphs
CDS	$O(n^3)$	see AT–free graphs
STEINER	$O(n^3)$	see AT–free graphs

For AT–free graphs with $\mathrm{diam}(G) > 3$ problem **CDS** can be solved in linear time (see [44]).

A.2.10 co–comparability graphs

co–comparability graphs $\subset$ AT–free

For a given co–comparability graph a co–comparability ordering can be computed in linear time (see [107]).

Problem	Time complexity	References
DS	$O(nm^2)$	[18], given a co–comparability ordering
DC–ex	$\mathbb{NP}$–c	[91]
TDS	$O(nm^2)$	[18], given a co–comparability ordering
CDS	$O(nm)$	[18], given a co–comparability ordering
WDS	$\mathbb{NP}$–c	[29]
WIDS	$O(n^{2.376})$	[18]
WIPDS	$O(n^2)$	[29], given a co–comparability ordering
WCDS	$\mathbb{NP}$–c	[29]
WTDS	$\mathbb{NP}$–c	[29]
STEINER	$\mathbb{P}$	[91]

A.2.11 comparability graphs

bipartite graphs $\subset$ comparability graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	see bipartite graphs
PDS	$\mathbb{NP}$–c	see bipartite graphs
DC–ex	$O(nm)$	[17]
DC–comp	$O(nm)$	[17]
IDS	$\mathbb{NP}$–c	see bipartite graphs
IPDS	$\mathbb{NP}$–c	see bipartite graphs
TDS	$\mathbb{NP}$–c	see bipartite graphs
TPDS	$\mathbb{NP}$–c	see bipartite graphs
CDS	$\mathbb{NP}$–c	see bipartite graphs
CPDS	$\mathbb{NP}$–c	see bipartite graphs
STEINER	$\mathbb{NP}$–c	[67]

A.2.12 convex bipartite

Problem	Time complexity	References
DS	$O(n^2)$	[14]
IDS	$O(n^2)$	[14]
TDS	$O(n^2)$	see **DS**, [92]
CDS	$O(n^4)$	[53]

A.2.13 convex–round graphs

Problem	Time complexity	References
DS	$O(n^3)$	[14]
IDS	$O(n^3)$	[14]
TDS	$O(n)$	[14], given a convex–round enumeration

A.2.14 distance–hereditary graphs

In [68] it is shown that the clique width of distance–hereditary graphs is bounded by three and that a 3–expression can be computed in the same time bound.

Problem	**Time complexity**	**References**
RDS	$O(n+m)$	Theorem 3.3.14, [109]
RDC–ex	$O(n+m)$	[50]
RDC–comp	$O(n+m)$	[50]
CRDS	$O(n+m)$	[10]
WDS	$O(n+m)$	see graph classes of bounded clique width
WPDS	$O(n+m)$	see graph classes of bounded clique width
WDC–ex	$O(n+m)$	see graph classes of bounded clique width
WDC–comp	$O(n+m)$	see graph classes of bounded clique width
WIDS	$O(n+m)$	see graph classes of bounded clique width
WIPDS	$O(n+m)$	see graph classes of bounded clique width
WCDS	$O(n+m)$	[126], see graph classes of bounded clique width
WCPDS	$O(n+m)$	see graph classes of bounded clique width
WTDS	$O(n+m)$	see graph classes of bounded clique width
WTPDS	$O(n+m)$	see graph classes of bounded clique width
STEINER	$O(n+m)$	[10], see graph classes of bounded clique width

A.2.15 DSP–graphs (graphs having a dominating shortest path)

co–comparability graphs $\subset$ AT–free graphs $\subset$ DSP–graphs

Problem	**Time complexity**	**References**
DS	$O(n^7)$	[89]
DC–ex	$\mathbb{NP}$–c	see co–comparability graphs
TDS	$O(n^7)$	[89]
WDS	$\mathbb{NP}$–c	see co–comparability graphs
WCDS	$\mathbb{NP}$–c	see co–comparability graphs
WTDS	$\mathbb{NP}$–c	see co–comparability graphs

A.2.16 doubly chordal graphs

doubly chordal graphs $=$ chordal graphs $\cap$ dually chordal graphs

Problem	Time complexity	References
RDS	$O(n+m)$	see dually chordal graphs
RDC–ex	$O(n+m)$	see dually chordal graphs
RDC–comp	$O(n+m)$	see dually chordal graphs
IDS	$O(n+m)$	see chordal graphs
CRDS	$O(n+m)$	[9]
WCPDS	$O(n+m)$	see chordal graphs

A.2.17 dually chordal graphs

Problem	Time complexity	References
RDS	$O(n+m)$	[9]
RDC–ex	$O(n+m)$	[51]
RDC–comp	$O(n+m)$	[51]
IDS	$\mathbb{NP}$–c	[9]
TDS	$O(n+m)$	see **DS**, [92]
CRDS	$O(m+\|E(G^2)\|)$	[9]

A.2.18 graph classes of bounded clique width

Let $\mathcal{C}$ be a graph class of bounded clique width such that for every graph $G \in \mathcal{C}$ a p–expression can be computed in $O(f(n,m))$ time.

Problem	Time complexity	References
WDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WPDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WDC–ex	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WDC–comp	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WIDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WIPDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WCDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WCPDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WTDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
WTPDS	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5
STEINER	$O(f(n,m))$	Theorem 3.1.3 and 3.1.5

A.2.19 homogeneously orderable graphs

In [13] an $O(n^3)$ recognition algorithm for homogeneously orderable graphs is given. A h–extremal ordering for a homogeneously orderable graph can be computed in the same time bound.

Problem	Time complexity	References
RDC–ex	$O(n^2)$	[55], given an h–extremal ordering
RDC–comp	$O(n^2)$	[55], given an h–extremal ordering
IDS	$\mathbb{NP}$–c	see dually–chordal graphs
CRDS	$O(n^2)$	[55], given an h–extremal ordering
STEINER	$O(\lvert E(G^2)\rvert)$	[13], given an h–extremal ordering

A.2.20 interval graphs

In [21] a linear time algorithm for recognizing interval graphs is given. In the same time bound an interval model can be constructed using PQ–trees.

Problem	Time complexity	References
RDS	$O(n+m)$	see dually chordal graphs
RDC–ex	$O(n+m)$	see dually chordal graphs
RDC–comp	$O(n+m)$	see dually chordal graphs
TDS	$O(n+m)$	[83]
WDS	$O(n)$	[30], given an interval model
WPDS	$O(n+m)$	[39]
WIDS	$O(n)$	[30], given an interval model
WIPDS	$O(n+m)$	[39]
WCDS	$O(n)$	[30], given an interval model
WCPDS	$O(n+m)$	[39]
WTDS	$O(n \log\log n)$	[30], given an interval model
	$O(n+m)$	[114]
WTPDS	$O(n+m)$	[39]
STEINER	$O(n^2)$	see strongly chordal graphs

A.2.21 line graphs

The edge version of domination can be thought of as the vertex version of the problem applied to line graphs.

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[128]
PDS	$\mathbb{NP}$–c	[102]
IDS	$\mathbb{NP}$–c	[128]
TDS	$\mathbb{NP}$–c	[102]

A.2.22 partial k–trees (for bounded k)

For a given k partial k–trees are exactly the graphs having tree width (twd) at most k. Since

$$cwd(G) \leq 2^{twd(G)+1} + 1 \qquad \text{(see [43])}$$

graphs of bounded tree width are of bounded clique width, too.

In [2] it is shown that for a given partial k–tree an embedding in a k–tree can be found in polynomial time.

Problem	Time complexity	References
DS	$O(n+m)$	[5], given a k–tree embedding
DC–comp	$\mathbb{P}$	[4]
IDS	$\mathbb{P}$	[4]
TDS	$\mathbb{P}$	[4]
CDS	$\mathbb{P}$	[4]
WDS	$\mathbb{P}$	see graph classes of bounded clique width
WPDS	$\mathbb{P}$	see graph classes of bounded clique width
WDC–ex	$\mathbb{P}$	see graph classes of bounded clique width
WDC–comp	$\mathbb{P}$	see graph classes of bounded clique width
WIDS	$\mathbb{P}$	see graph classes of bounded clique width
WIPDS	$\mathbb{P}$	see graph classes of bounded clique width
WCDS	$\mathbb{P}$	see graph classes of bounded clique width
WCPDS	$\mathbb{P}$	see graph classes of bounded clique width
WTDS	$\mathbb{P}$	see graph classes of bounded clique width
WTPDS	$\mathbb{P}$	see graph classes of bounded clique width
STEINER	$\mathbb{P}$	see graph classes of bounded clique width

A.2.23 permutation graphs

permutation graphs $\subset$ trapezoid graphs

In [107] a linear time recognition algorithm for permutation graphs is given. A permutation diagram for a permutation graph can be computed in the same time bound.

Problem	Time complexity	References
DS	$O(n)$	[35], given a permutation diagram
DC–ex	$O(n)$	[16, 17, 81], given a permutation diagram
DC–comp	$O(n)$	[17, 81], given a permutation diagram
IDS	$O(n \log n)$	[3], given a permutation diagram
TDS	$O(n)$	see **DS**, [92], given a permutation diagram
CDS	$O(n)$	[81], given a permutation diagram
WDS	$O(n+m)$	[113]
WPDS	$O(n \log n)$	see trapezoid graphs
WDC–ex	$O(n \log n)$	[119], given a permutation diagram
WDC–comp	$O(n \log n)$	[119], given a permutation diagram
WIDS	$O(n+m)$	[99]
WCDS	$O(n+m)$	[94]
STEINER	$O(n \log n)$	[81], given a permutation diagram

A.2.24 planar graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[82], see planar bipartite graphs
PDS	$\mathbb{NP}$–c	[86]
IDS	$\mathbb{NP}$–c	see planar bipartite graphs
CDS	$\mathbb{NP}$–c	see planar bipartite graphs
STEINER	$\mathbb{NP}$–c	[66], see planar bipartite graphs

A.2.25 planar bipartite graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[132]
IDS	$\mathbb{NP}$–c	[132]
CDS	$\mathbb{NP}$–c	[124]
WDC–ex	$O(n+m)$	see bipartite graphs
WDC–comp	$O(n+m)$	see bipartite graphs
STEINER	$\mathbb{NP}$–c	[124]

In [132] it is shown that **DS** and **IDS** are $\mathbb{NP}$–c for the subclass $\mathcal{L}$ of planar bipartite graphs. Hereby, G belongs to $\mathcal{L}$ if the following conditions are fulfilled:

- G is planar,
- G is bipartite,
- G has maximum degree 3,
- G has the girth $g(G) \geq k$, where k is fixed.

A.2.26 k–polygon graphs for fixed $k \geq 3$

In [59] an $O(4^k n^2)$ time algorithm recognizing k–polygon graphs (for fixed $k \geq 3$) is given.

Problem	Time complexity	References
DS	$O(n^{4k^2+3})$	[59, 58], given a k–polygon representation
DC–comp	$\mathbb{P}$	[84]
IDS	$O(2^{k^2} n^{4k-4})$	[58], given a k–polygon representation
TDS	$O(n^{4k^2+3})$	see **DS**, [92], given a k–polygon representation
CDS	$O(n^{4k^2+3})$	[58], given a k–polygon representation

A.2.27 series–parallel graphs = partial 2–trees

Problem	Time complexity	References
DS	$O(n)$	[93]
IDS	$O(n$	[112], see partial k–trees
TDS	$O(n)$	[112], see partial k–trees
WDS	$\mathbb{P}$	see partial k–trees
WPDS	$O(n)$	[125, 130]
WDC–ex	$\mathbb{P}$	see partial k–trees
WDC–comp	$\mathbb{P}$	see partial k–trees
WIDS	$\mathbb{P}$	see partial k–trees
WIPDS	$O(n)$	[125]
WCDS	$O(n)$	[124]
WCPDS	$O(n)$	[125]
WTDS	$\mathbb{P}$	see partial k–trees
WTPDS	$O(n)$	[125]
STEINER	$O(n)$	[123]

A.2.28 split graphs

split graphs $\subset$ chordal graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[45]
DC–comp	$\mathbb{NP}$–c	see chordal
RDC–ex	$O(nm)$	see chordal
IDS	$O(n+m)$	see chordal
TDS	$\mathbb{NP}$–c	[47]
CDS	$\mathbb{NP}$–c	[124]
WPDS	$O(n+m)$	[38]
WIPDS	$O(n+m)$	[38]
WCPDS	$O(n+m)$	[38]
WTPDS	$O(n+m)$	[38]
STEINER	$\mathbb{NP}$–c	[124]

A.2.29 strongly chordal

strongly chordal $=$ hereditary dually chordal $\subset$ dually chordal

For a given graph G one can find a simple (resp. strong) elimination ordering of G in $O(\min\{m \log n, n^2\})$ (resp. $O(n^3)$) time (see [23]).

Problem	Time complexity	References
RDS	$O(n+m)$	see dually chordal
	$O(n+m)$	[30], given a simple elimination ordering
RDC–ex	$O(n+m)$	see dually chordal
RDC–comp	$O(n+m)$	see dually chordal
TDS	$O(n+m)$	see dually chordal
	$O(n+m)$	[28], given a simple elimination ordering
CRDS	$O(n+m)$	[30], given a simple elimination ordering
WDS	$O(n+m)$	[63], given a strong elimination ordering
WIDS	$O(n+m)$	[63], given a strong elimination ordering
WCPDS	$O(n+m)$	see chordal graphs
STEINER	$O(n^2)$	[124], given a strong or simple elimination ordering

A.2.30 trapezoid graphs

trapezoid graphs $\subset$ co–comparability graphs

Problem	Time complexity	References
DC–ex	$O(n+m)$	[85], given a trapezoid order
DC–comp	$O(n+m)$	[85], given a trapezoid order
TDS	$O(nm)$	[95]
CDS	$O(n)$	[85], given a trapezoid diagram
WDS	$O(nm)$	[95]
WPDS	$O(n \log n)$	[96], given a trapezoid diagram
WDC–ex	$O(m \log^4 n)$	[1]
WDC–comp	$O(m \log^4 n)$	[1]
WIDS	$O(n \log n)$	[96], given a trapezoid diagram
WIPDS	$O(n^2)$	see co–comparability graphs
WCDS	$O(m + n \log n)$	[117], given a trapezoid diagram
STEINER	$\mathbb{P}$	see co–comparability graphs

A.2.31 trees

trees $\subset$ distance–hereditary graphs

Problem	Time complexity	References
DS	$O(n)$	[33]
RDS	$O(n)$	[116]
RDC–ex	$O(n)$	trivial
RDC–comp	$O(n)$	trivial
CRDS	$O(n)$	see distance–hereditary graphs
WDS	$O(n)$	[110]
WPDS	$O(n)$	[129]
WDC–ex	$O(n)$	trivial
WDC–comp	$O(n)$	trivial
WIDS	$O(n)$	see distance–hereditary graphs
WIPDS	$O(n)$	[125]
WCDS	$O(n)$	see distance–hereditary graphs
WCPDS	$O(n)$	[125]
WTDS	$O(n)$	see distance–hereditary graphs
WTPDS	$O(n)$	[125]
STEINER	$O(n)$	trivial

A.2.32 undirected path graphs

undirected path graphs $\subset$ chordal graphs

Problem	Time complexity	References
DS	$\mathbb{NP}$–c	[15]
DC–ex	$\mathbb{P}$	[88]
DC–comp	$\mathbb{P}$	[88]
RDC–ex	$O(nm)$	see chordal graphs
IDS	$O(n+m)$	see chordal graphs
TDS	$\mathbb{NP}$–c	[98]
CDS	$\mathbb{NP}$–c	[47]
WCPDS	$O(n+m)$	see chordal graphs

A.2.33 weakly chordal graphs

split graphs $\subset$ chordal graphs $\subset$ weakly chordal graphs

Problem	**Time complexity**	**References**
DS	$\mathbb{NP}$–c	see split graphs
PDS	$\mathbb{NP}$–c	see chordal graphs
DC–ex	$\mathbb{NP}$–c	[17]
DC–comp	$\mathbb{NP}$–c	[17]
IPDS	$\mathbb{NP}$–c	see chordal graphs
TDS	$\mathbb{NP}$–c	see split graphs
TPDS	$\mathbb{NP}$–c	see chordal graphs
CDS	$\mathbb{NP}$–c	see split graphs
WIDS	$\mathbb{NP}$–c	see chordal graphs
STEINER	$\mathbb{NP}$–c	see split graphs

Bibliography

[1] K. ARVIND, H. BREU, M.-S. CHANG, D.G. KIRKPATRICK, F.-Y. LEE, Y.D. LIANG, K. MADHUKAR, C. PANDU RANGAN, A. SRINIVASAN, Efficient algorithms in cocomparability graphs and trapezoid graphs, *Manuscript*, 1999.

[2] S. ARNBORG, D.G. CORNEIL, A. PROSKUROWSKI, Complexity of finding embeddings in a k–tree, *SIAM J. Alg. Discr. Methods* 8 (1987), 277–284.

[3] M.J. ATALLAH, S.R. KOSARAJU, An efficient algorithm for maxdominance, with applications, *Algorithmica* 4 (1989), 221–236.

[4] S. ARNBORG, J. LAGERGREN, D. SEESE, Easy problems for tree-decomposable graphs, *J. Algorithms* 12 (1991), 308–340.

[5] S. ARNBORG, A. PROSKUROWSKI, Linear time algorithms for $\mathbb{NP}$-hard problems restricted to partial k-trees, *Discr. Appl. Math.* 23 (1989), 11–24.

[6] K. ARVIND, C.P. REGAN, Connected domination and Steiner set on weighted permutation graphs, *Inform. Proces. Lett.* 41 (1992), 215–220.

[7] A.A. BERTOSSI, Total domination in interval graphs, *Inform. Proces. Lett.* 23 (1986), 131-134.

[8] D.W. BANGE, A.E. BARKAUSKAS, P.J. SLATER, Efficient dominating sets in graphs, in: *Applications of Discrete Mathematics*, R.D. Ringeisen, F.S. Roberts, (eds.), SIAM, Philadelphia (1988), 189–199.

[9] A. BRANDSTÄDT, V.D. CHEPOI, F.F. DRAGAN, The algorithmic use of hypertree structure and maximum neighborhood orderings, *Discr. Appl. Math.* 82(1-3) (1998), 43–77.

[10] A. BRANDSTÄDT, F.F. DRAGAN, A linear–time algorithm for connected r–domination and Steiner Tree on distance–hereditary graphs, *Networks* 31 (1998), 177-182.

[11] A. BRANDSTÄDT, F.F. DRAGAN, On the clique width of graph classes defined by three forbidden P_4 extensions, *Manuscript*, 2001.

[12] A. BRANDSTÄDT, F.F. DRAGAN, V.D. CHEPOI, V.I. VOLOSHIN, Dually chordal graphs, *SIAM J. Discrete Math.* 11 (1998), 437–455.

[13] A. BRANDSTÄDT, F.F. DRAGAN, F. NICOLAI, Homogeneously orderable graphs, *Theor. Comp. Sc.* 172 (1997), 209–232.

[14] J. BANG–JENSEN, J. HUANG, G. MACGILLIVRAY, A. YEO, Domination in convex bipartite and convex–round graphs, *Manuscript*, 1999.

[15] K.S. BOOTH, J.H. JOHNSON, Dominating sets in chordal graphs, *SIAM J. Comp.* 11(1) (1982), 191–199.

[16] A. BRANDSTÄDT, D. KRATSCH, On the restriction of some $\mathbb{NP}$–complete problems to permutation graphs, in: *Proceedings of FCT'85, Springer, Lecture Notes in Computer Science* 199 (1985) 53–62.

[17] A. BRANDSTÄDT, D. KRATSCH, Domination problems on permutation and other graphs, *Theor. Comp. Sc.* 54 (1987), 181–198.

[18] H. BREU, D.G. KIRKPATRICK, Algorithms for dominating and Steiner set problems in cocomparability graphs, *Manuscript*, 1993.

[19] H. BROERSMA, T. KLOKS, D. KRATSCH, H. MÜLLER, Independent sets in asteroidal triple–free graphs *Proceedings of ICALP'97, Springer, Lecture Notes in Computer Science* 1256 (1997), 760–770.

[20] A. BRANDSTÄDT, T. KLEMBT, V.B. LE, S. MAHFUD, T. SZYMCZAK, On the structure and clique width of graph classes defined by two forbidden four-vertex graphs, *Manuscript*, 2001.

[21] K.S. BOOTH, G.S. LUEKER, Testing for consecutive ones property, interval graphs and graph planarity using PQ–tree algorithm, *J. Comput. System Sci.* 13 (1976), 335–379.

[22] A. BRANDSTÄDT, V.V. LOZIN, On the linear structure and clique–width of bipartite permutation graphs, *Rutcor Research Report* RRR 29–2001, RUTCOR Rutgers Center for Operations Research, Rutgers University, New Jersey, 2001.

[23] A. BRANDSTÄDT, V.B. LE, J. SPINRAD, Graph classes: A survey, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.

[24] H.-J. BANDELT, H.M. MULDER, Distance–hereditary graphs, *J. Comb. Theory B* 41 (1986), 182–208.

[25] T. BEYER, A. PROSKUROWSKI, S. HEDETNIEMI, S. MITCHELL, Independent domination in trees, in: *Proceedings of the 8th Southeastern Conference on Combinatorics, Graph Theory and computing*, Utilitas Mathematica, Winnipeg (1977), 321–328.

[26] H. BALAKRISHNAN, R. RAJARAMAN, C. PANDU RANGAN, Connected domination and Steiner set on asteroidal triple–free graphs, *Proceedings of WADS'93*, Lecture Notes in Computer Science, Vol. 709, Springer, Berlin, 1993, 131–141.

[27] G. BASCÓ, Z. TUZA, Dominating cliques in P_5-free graphs, *Period. Math. Hungar.* 21, 1990, 303–308.

[28] M.S. CHANG, Labelling algorithms for domination problems in sunfree chordal graphs, *Discr. Appl. Math.* 22 (1988/89), 21–34.

[29] M.S. CHANG, Weighted domination on cocomparability graphs, In: *Proceedings ISAAC'95, Springer, Lecture Notes in Computer Science* 1004 (1995), 122–131, *Discr. Appl. Math.* 80(2-3) (1997), 135–148.

[30] M.S. CHANG, Efficient algorithms for the domination problems on interval and circular–arc graphs, *SIAM J. Comp.* 27 (1998), 1671–1694.

[31] B. COURCELLE, The expression of graph properties and graph transformations in monadic second-order logic, In: *Handbook of Graph Grammars and Computing by Graph Transformations*, volume 1, chapter 5, 313–400, G. Rozenberg (ed.), World Scientific, Singapore, 1997.

[32] B. COURCELLE, J. ENGELFRIET, G. ROZENBERG, Handle–rewriting hypergraph grammars, *J. Comput. System Sci.* 46 (1993), 218–270.

[33] E.J. COCKAYNE, S.E. GOODMAN, S.T. HEDETNIEMI, A linear algorithm for the domination number of a tree, *Inform. Proces. Lett.* 4 (1975), 41–44.

[34] A. COURNIER, M. HABIB, A simple linear algorithm to build modular decomposition trees, *Technical Report* LIRMM 94-063, Montpellier 1994.

[35] H.S. CHAO, F.R. HSU, R.C.T. LEE, An optimal algorithm for finding the minimum cardinality dominating set on permutation graphs, *Discr. Appl. Math.* 102 (2000), 159–173.

[36] M. CHEIN, M. HABIB, M.C. MAURER, Partitive hypergraphs, *Discr. Math.* 37 (1981), 35–50.

[37] D. CORNEIL, J.M. KEIL, A dynamic programming approach the dominating set problem on k-trees, *SIAM J. Algebraic Discrete Math.* 8 (1987), 534–543.

[38] M.S. CHANG, Y.C. LIU, Polynomial algorithms for the weighted perfect domination problems on chordal graphs and split graphs, *Inform. Proces. Lett.* 48 (1993), 205–210.

[39] M.S. CHANG, Y.C. LIU, Polynomial algorithms for the weighted perfect domination problems on interval graphs, *Manuscript*, 1993.

[40] B. COURCELLE, A. MAKOWSKY, U. ROTICS, Linear time solvable optimization problems on graphs of bounded clique–width, *Theory Comput. Systems* 33 (2000), 125–150.

[41] G.J. CHANG, G.L. NEMHAUSER, R–domination of block graphs, *Operations Res. Lett.* 1 (1982), 214–218.

[42] G.J. CHANG, G.L. NEMHAUSER, The k–domination and k–stability problems on sun–free chordal graphs, *SIAM J. Algebraic and Discrete Methods* 5 (1984), 332–345.

[43] B. COURCELLE, S. OLARIU, Upper bounds to the clique–width of graphs, *Discr. Appl. Math.* 101 (2000), 77–114.

[44] D.G. CORNEIL, S. OLARIU, L. STEWART, A linear time algorithm to compute dominating pairs in asteroidal triple–free graphs, in: *Proceedings of ICALP'95, Springer, Lecture Notes in Computer Science* 944 (1995), 292–302.

[45] D.G. CORNEIL, Y. PERL, Clustering and domination in perfect graphs, *Discr. Appl. Math.* 9 (1984), 27–39.

[46] G.J. CHANG, C. PANDU–RANGAN, S.R. COORG, Weighted independent perfect domination on cocomparability graphs, *Discr. Appl. Math.* 63 (1995), 215–222.

[47] D.G. CORNEIL, L. STEWART, Dominating sets in perfect graphs, *Discr. Math.* 86 (1990), 145–164.

[48] C.J. COLBOURN, L. STEWART, Permutation graphs: connected domination and Steiner trees, *Discr. Math.* 86 (1990), 179–189.

[49] A.K. DEWDNEY, Fast Turing reductions between problems in NP 4, *Technical Report* 71, University of Western Ontario, 1981.

[50] F.F. DRAGAN, Dominating cliques in distance–hereditary graphs, in: *Proceedings of "Algorithm Theory — SWAT94"*, 4th Scandinavian Workshop on Algorithm Theory, Aarhus, Denmark, *Springer, Lecture Notes in Computer Science* 824 (1994), 370–381.

[51] F.F. DRAGAN, A. BRANDSTÄDT, r–Dominating cliques in graphs with hypertree structure, *Discr. Math.* 162 (1996), 93–108.

[52] A. D'ATRI, M. MOSCARINI, Distance–hereditary graphs, Steiner trees, and connected domination, *SIAM J. Comput.* 17(3) (1988), 521–538.

[53] P. DAMASCHKE, H. MÜLLER, D. KRATSCH, Domination in convex and chordal bipartite graphs, *Inform. Proces. Lett.* 36 (1990), 231-236.

[54] A. D'ATRI, M. MOSCARINI, A. SASSANO, The Steiner tree problem and homogeneous sets, MFCS '88, *Springer, Lecture Notes in Computer Science* 324, 249–261.

[55] F.F. DRAGAN, F. NICOLAI, r–Domination problems on homogeneously orderable graphs, *Networks* 30 (1997), 121–131.

[56] T. EMDEN–WEINERT, S. HOUGARDY, B. KREUTER The complexity of some problems on very sparse graphs, *Technical Report* , 1997.

[57] P. ERDŐS, M. SIMONOVITS, Compactness results in extremal graph theory, *Combinat.* 2 (1982), 275–288.

[58] E.S. ELMALLAH, L.K. STEWART, Domination in polygon graphs, *Congr. Numer.* 77 (1990), 63–76.

[59] E.S. ELMALLAH, L.K. STEWART, Independence and domination in polygon graphs, *Discr. Appl. Math.* 44 (1993), 65–77.

[60] E.S. ELMALLAH, L.K. STEWART, Polygon graph recognition, *J. Algorithms* 26 (1998), 101–140.

[61] M. FARBER, Applications of linear programming duality to problems involving independence and domination, *Ph.D. Theses*, Simon Fraser University, TR–81–13, 1981.

[62] M. FARBER, Independent domination in chordal graphs, *Oper. Res. Lett.* 1 (1982), 134–138.

[63] M. FARBER, Domination, independent domination, and duality in strongly chordal graphs, *Discr. Appl. Math.* 7 (1984), 115–130.

[64] R. FAUDREE, E. FLANDRIN, Z. RYJÁCEK, Claw–free graphs — a survey, *Discr. Math.* 164 (1997), 87–147.

[65] M. FARBER, M. KEIL, Domination in permutation graphs, *J. Algorithms* 6 (1985), 309–321.

[66] M.R. GAREY, D.S. JOHNSON, The rectilinear Steiner tree problem is $\mathbb{NP}$–complete, *SIAM J. Appl. Math.* 32 (1977), 826-834.

[67] M.R. GAREY, D.S. JOHNSON, Computers and Intractability: A guide to the theory of $\mathbb{NP}$–completeness, W.H. Freeman, 1979.

[68] M.C. GOLUMBIC, U. ROTICS, On the Clique–Width of some Perfect Graph Classes, *International Journal of Foundations of Computer Science* 11(3) (2000), 423–443.

[69] E. HOWORKA, A characterization of distance–hereditary graphs, *Quart. J. Math. Oxford* Ser. 2, 28 (1977), 417-420.

[70] S.Y. HSIEH, An efficient parallel algorithm for the efficient domination problem on distance–hereditary graphs, in: *Proceedings of the 4th Austrian–Hungarian Workshop on Distributed and Parallel Systems DAPSYS'98* (1998).

[71] S.F. HWANG, G.J. CHANG, The edge domination problem, *Discuss. Math. – Graph Theory* 15 (1995), 51–57.

[72] J.D. HORTON, K. KILAKOS, Minimum edge dominating sets, *SIAM J. Disc. Math.* 6(3) (1993), 375–387.

[73] H. HEMPEL, D. KRATSCH, On claw–free asteroidal triple–free graphs, *Manuscript*, 1999.

[74] S.T. HEDETNIEMI, R. LASKAR, Recent results and open problems in domination theory, in: *Applications of Discrete Mathematics*, R.O. Ringeisen and F.S. Roberts (eds.), SIAM, Philadelphia (1988) 205–218.

[75] S.T. HEDETNIEMI, R. LASKAR, (Eds.), Topics on domination, *Ann. Discr. Math.* 48 (1991).

[76] S.T. HEDETNIEMI, R. LASKAR, J. PFAFF, A linear algorithm for finding a minimum dominating set in a cactus, *Discr. Appl. Math.* 13 (1986), 287–292.

[77] P.L. HAMMER, F. MAFFRAY, Completely separable graphs, *Discr. Appl. Math.* 27 (1990), 85–99.

[78] S.T. HEDETNIEMI, A.A. MCRAE, D.A. PARKS, Complexity results, (Chapter 9), in: *Domination in Graphs: Advanced Topics*, Pure and Applied Mathematics, A series of Monographs and Textbooks, Vol. 209, T.W. Haynes, S.T. Hedetniemi, P.J. Slater, (eds.), Marcel Dekker, 1998, 233–269.

[79] F.K. HWANG, D.S. RICHARDS, P. WINTER, The Steiner tree problem, *Monograph, Annals of Discrete Mathematics* 53, North–Holland, 1992.

[80] W.-L. HSU, K.-H. TSAI, Linear time algorithms on circular–arc graphs, *Inform. Proces. Lett.* 40(3) (1991), 123–129.

[81] O.H. IBARRA, Q. ZHENG, Some efficient algorithms for permutation graphs, *J. Algorithms* 16 (1994), 453–469.

[82] D.D. JOHNSON, The $\mathbb{NP}$-completeness column: an ongoing guide, *J. Algorithms* 5 (1984), 147–160.

[83] J.M. KEIL, Total domination in interval graphs, *Inform. Proces. Lett.* 22 (1986), 171-174.

[84] J.M. KEIL, The complexity of domination problems in circle graphs, *Discr. Appl. Math.* 42 (1993), 51–63.

[85] E. KÖHLER, Connected domination and dominating clique in trapezoid graphs, *Discr. Appl. Math.* 99 (2000), 91–110.

[86] J. KRATOCHVIL, Perfect codes in general graphs, *Monograph*, Academia Praha, 1991. 1998, 191–231.

[87] D. KRATSCH, Algorithms, (Chapter 8), in: *Domination in Graphs: Advanced Topics*, Pure and Applied Mathematics, A series of Monographs and Textbooks, Vol. 209, T.W. Haynes, S.T. Hedetniemi, P.J. Slater, (eds.), Marcel Dekker, 1998, 191–231.

[88] D. KRATSCH, Finding dominating cliques efficiently in strongly chordal graphs and undirected path graphs, *Annals of Discr. Math.* 48, Topics on Domination (S.L. Hedetniemi, R.C. Laskar, eds.) (1990), 225-238. 1998, 191–231.

[89] D. KRATSCH, Domination and total domination on asteroidal triple–free graphs, *Discr. Appl. Math.* 99 (2000), 111–123.

[90] D. KRATSCH, P. DAMASCHKE, A. LUBIW, Dominating cliques in chordal graphs, *Discr. Math.* 128 (1994), 269–275.

[91] D. KRATSCH, L. STEWART, Domination on cocomparability graphs, *SIAM J. Discr. Math.* 6 (1993), 400–417.

[92] D. KRATSCH, L. STEWART, Total domination and transformation, *Inform. Proces. Lett.* 63(3) (1997), 167–170.

[93] T. KIKUNO, N. YOSHIDA, Y. KOKUDA, A linear algorithm for the domination number of series–parallel graphs, *Discr. Appl. Math.* 5 (1983), 299–311.

[94] Y.D. LIANG, Connected domination and Steiner trees in permutation graphs, in: *ICS'94, 1994 Internat. Comput. Symp.* (1994), 13–18.

[95] Y.D. LIANG, Dominations in trapezoid graphs, *Inform. Proces. Lett.* 52(6) (1994), 309–315.

[96] Y.-L. LIN, Fast Algorithms for Independent Domination and Efficient Domination in Trapezoid Graphs, *Manuscript*, 1999.

[97] R. LASKAR, J. PFAFF, Domination and irredundance in split graphs, *Technical Report* 430, Clemson University, 1983.

[98] R. LASKAR, J. PFAFF, S.M. HEDETNIEMI, S.T. HEDETNIEMI, On the algorithm complexity of total domination, *SIAM J. Algebraic and Discrete Methods* 5 (1984), 420-425.

[99] Y.D. LIANG, C. RHEE, Linear algorithms for two independent set problems in permutation graphs, in: *Proc. of 22nd Computer Science Conference* (1994), 90–93.

[100] C.L. LU, C.Y. TANG, Efficient domination on bipartite graphs, *Manuscript*, 1996.

[101] R.M. MCCONNELL, Linear–time recognition of circular–arc graphs, *Manuscript*, 2001.

[102] A.A. MCRAE, Generalizing $\mathbb{NP}$–completeness proofs for bipartite and chordal graphs, *Ph.D. Thesis*, Clemson University, 1994.

[103] H. MÜLLER, A. BRANDSTÄDT, The $\mathbb{NP}$–completeness of Steiner Tree and Dominating Set for chordal bipartite graphs, *Theor. Comp. Sc.* 53 (1987), 257–265.

[104] S. MITCHELL, S. HEDETNIEMI, Edge domination in trees, *Proc. Eighth Southeastern Conf. on Combinatorics, Graph Theory and Computing*, Utilitas Mathematica, Winnipeg, 1977.

[105] J.A. MAKOWSKY, U. ROTICS, On the clique-width of graphs with few P_4's, *Int. J. Foundations of Computer Science* 10 (1999), 329–348.

[106] R. MCCONNELL, J. SPINRAD, Linear–time modular decomposition and efficient transitive orientation of comparability graphs, *Fifth Annual ACM–SIAM Symposium of Discrete Algorithms* (1994), 536–545.

[107] R. MCCONNELL, J. SPINRAD, Modular decomposition and transitive orientation, *Discr. Math.* 201 (1999), 189–241.

[108] F. NICOLAI, T. SZYMCZAK, r–Domination problems on trees and their homogeneous extensions, *International Journal of Mathematical Algorithms* 1 (1999), 53–79.

[109] F. NICOLAI, T. SZYMCZAK, Homogeneous sets and domination: A linear time algorithm for distance–hereditary graphs, *Networks* 37(3) (2001), 117–128.

[110] K.S. NATARAJAN, L.J. WHITE, Optimum domination in weighted trees, *Inform. Proces. Lett.* 7(6) (1978), 261-265.

[111] J. PFAFF, R. LASKAR, S.T. HEDETNIEMI, $\mathbb{NP}$–completeness of total and connected domination and irredundance for bipartite graphs, *Technical Report* 428, Clemson University, 1983.

[112] J. PFAFF, R. LASKAR, S.T. HEDETNIEMI, Linear algorithms for independent domination and total domination in series–parallel graphs, *Technical Report* 441, Clemson University, 1984.

[113] C. RHEE, Y.D. LIANG, S.K. DHALL, S. LAKSHMIVARAHAN, An $O(n+m)$–time algorithm for finding a minimum–weight dominating set in a permutation graph, *SIAM J. Comp.* 25 (1996), 404–419.

[114] G. RAMALINGAM, C.P. RANGAN, A unified approach to domination on interval graphs, *Inform. Proces. Lett.* 27 (1988), 271-274.

[115] U. SCHÖNING, Logik für Informatiker, 4. Auflage, *Spektrum, Akademischer Verlag*, 1995.

[116] P.J. SLATER, R–Domination in graphs, *Journal of the Association for Computing Machinery* 23(3) (1976), 446–450.

[117] A. SRINIVASAN, M.S. CHANG, K. MADHUKAR, C.P. RANGAN, Efficient algorithms for the weighted domination problems on trapezoid graphs, *Manuscript*, 1996.

[118] A. SRINIVASAN, K. MADHUKAR, P. NAGAVAMSI, C.P. RANGAN, M.-S. CHANG, Edge domination in bipartite permutation graphs and cotriangulated graphs, *Inform. Proces. Lett.* 56 (1995), 165–171.

[119] A. SRINIVASAN, C.P. RANGAN, Efficient algorithms for the minimum weighted dominating clique problem on permutation graphs, *Theor. Comp. Sc.* 91 (1991), 1–21.

[120] J.A. TELLE, Complexity of domination–type problems in graphs, *Nordic Journal of Computing* 1 (1994), 157–171.

[121] K. TSAI, W.L. HSU, Fast algorithms for the dominating set problem on permutation graphs, *Algorithmica* 9 (1993), 109–117.

[122] J.A. WALD, C.J. COLBOURN, Steiner trees in outerplanar graphs, in: *Proceedings 13th Southeastern Conference on Combinatorics, Graph Theory, and Computing* (1982), 15–22.

[123] J.A. WALD, C.J. COLBOURN, Steiner trees, partial 2–trees, and minimum IFI networks, *Networks* 13 (1983), 159–167.

[124] K. WHITE, M. FARBER, W. PULLEYBLANK, Steiner trees, connected domination and strongly chordal graphs, *Networks* 15 (1985), 109–124.

[125] C.C. YEN, Algorithmic aspects of perfect domination, *Ph.D. Thesis*, Institute of Computer Science, National Tsing Hua University, Taiwan, ROC, 1992.

[126] H.-G. YEH, G.J. CHANG, Weighted connected domination and Steiner trees in distance–hereditary graphs, *Discr. Appl. Math.* 87 (1998), 245–253.

[127] H.-G. YEH, G.J. CHANG, Weighted connected k–domination and weighted k–dominating clique in distance–hereditary graphs, *Theor. Comp. Sc.* 263 (2001), 3–8.

[128] M. YANNAKAKIS, F. GAVRIL, Edge dominating sets in graphs, *SIAM J. Appl. Math.* 38(3) (1980), 364–372.

[129] C.C. YEN, R.C.T. LEE, The weighted perfect domination problem, *Inform. Proces. Lett.* 35 (1990), 295–299.

[130] C.C. YEN, R.C.T. LEE, A linear time algorithm to solve the weighted perfect domination problem in series–parallel graphs, *European J. Oper. Res.*, to appear.

[131] C.C. YEN, R.C.T. LEE, The weighted perfect domination problem and its variants, *Discr. Appl. Math.* 66 (1996), 147–160.

[132] I.E. ZVEROVICH, V.E. ZVEROVICH, An induced subgraph characterization of domination perfect graphs, *Journal of Graph Theory* 20 (3) (1995), 375–395.

Index

Thesen

1. Viele Probleme, die sich mit dem strategischen Plazieren von irgendwelchen Objekten in einem Netzwerk beschäftigen, lassen sich als Dominationsprobleme in gewissen Graphen beschreiben. Im einfachsten Fall, dem sogenannten Problem MINIMUM DOMINATING SET sucht man zu einem gegebenen Graphen $G = (V, E)$ eine kleinste Teilmenge D von V so, daß für jeden Knoten v aus V gilt: $v \in D$ oder v hat einen Nachbarn in D. Durch die Betrachtung von konkreten Anwendungen wurden mehrere Varianten dieses Problems formuliert und algorithmisch untersucht. Zu erwähnen ist hierbei u.a. das Steinerbaumproblem (STEINER TREE), ein Spezialfall des Problems MINIMUM CONNECTED r–DOMINATING SET, welches im VLSI–Design verwendet wird.

2. Leider ist MINIMUM DOMINATING SET, wie auch fast alle Varianten hiervon, ein $\mathbb{NP}$–vollständiges Problem. Daher ist es interessant, die Komplexität dieses Problems für spezielle Graphenklassen zu untersuchen. Insbesondere möchte man für möglichst große Klassen effiziente Algorithmen entwickeln.

3. Graphenklassen mit beschränkter Cliquenweite besitzen in der algorithmischen Graphentheorie eine große Bedeutung. Für solche Klassen existiert ein gemeinsamer Zugang, mit dem sich viele Graphenprobleme (sozusagen gleichzeitig) effizient lösen lassen. Voraussetzung hierfür ist u.a., daß sich das gegebene Problem in monadischer Logik zweiter Ordung beschreiben läßt. MINIMUM DOMINATING SET ist ein solches Problem, das Problem MINIMUM r–DOMINATING SET dagegen nicht.

4. Bis auf vier ausnahmen läßt sich für jede Graphenklasse, die durch zwei verbotene vier–Knoten Graphen definiert ist, angeben, ob die Cliquenweite durch eine Konstante beschränkt ist, oder nicht (siehe Figure 3.1 in der Dissertation). Für die folgenden Klassen ist dieses Problem noch offen:

   - $\mathcal{C}_1 = (K_4, 2K_2)$-freie Graphen,
   - $\mathcal{C}_2 = (K_4$, co–diamond)-freie Graphen,
   - $\mathcal{C}_3 =$ co–$\mathcal{C}_1 = (4K_1, C_4)$-freie Graphen,
   - $\mathcal{C}_3 =$ co–$\mathcal{C}_2 = (4K_1$, diamond)-freie Graphen.

5. Es sei $\mathcal{F}$ eine Menge von Graphen, die höchstens vier Knoten enthalten. Für $\mathcal{F}$–freie Graphen läßt sich MINIMUM DOMINATING SET entweder in Polynomialzeit lösen,

oder $\mathcal{F}$ ist eine Oberklasse der folgenden Klassen und MINIMUM DOMINATING SET ist $\mathbb{NP}$–vollständig:

- $(C_4, 2K_2)$-freie Graphen,
- (claw, C_4, diamond, K_4)-freie Graphen,
- (K_3, C_4)-freie Graphen.

6. Die Graphenklasse $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ ist eine Teilklasse der homogen geordneten Graphen, welche die distanz–erblichen Graphen enthält, und unbeschränkte Cliquenweite besitzt.

   (a) Die Erkennung, ob ein Graph Element der Klasse $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ ist, läßt sich in Zeit $O(|V||E|)$ durchführen.

   (b) Die Probleme MINIMUM r–DOMINATING SET und TOTAL DOMINATING SET lassen sich für Graphen aus $\Gamma_{\{\mathrm{PV,HExt}\}}(K_1)$ in Zeit $O(|V||E|)$ lösen. Für die Teilklasse der distanz–erblichen Graphen verringert sich die Zeitschranke auf Linearzeit.

7. In den letzten Jahren wurde am Institut für Theoretische Informatik der Universität Rostock ein Informationssystem über Graphenklasseninklusionen (ISGCI) entwickelt, welches im Internet unter der URL

   `http://www.informatik.uni-rostock.de/~gdb/isgci/Isgci.html`

   erreichbar ist. Die im Appendix A der Dissertation angegebene Übersicht der Komplexiät von 27 Dominationsproblemen für verschiedene Graphenklassen kann als Ausgangspunkt dienen, dieses System mit Informationen über graphentheoretische Probleme zu erweitern.

Tabellarischer Lebenslauf

Angaben zur Person

Name	Thomas Szymczak
Geburtsdatum, Geburtsort	13.03.2001, Dinslaken (NRW)
Staatsangehörigkeit	deutsch
Familienstand	ledig

Schulausbildung / Studium

1977-1981	Grundschule in Dinslaken-Hiesfeld
1981-1990	Otto-Hahn-Gymnasium in Dinslaken *Abschluß:* Allgemeine Hochschulreife
1991-1995	Studium der Mathematik mit Nebenfach Informatik an der Gerhard-Mercator-Universität – Gesamthochschule Duisburg *Abschluß:* Diplom-Mathematiker

Grundwehrdienst

1990-1991	Ausbildung als Kfz- und Panzerschlosser in Varel und Wesel

Weitere Tätigkeiten

1993-1994	Entwicklung eines Übungsbuches zur Analysis für den Vieweg Verlag
1995	DFG-Forschungsstudent bei Prof. Dr. A. Brandstädt *Thema:* Parallele und sequentielle Algorithmen für Probleme auf Hypergraphen und Graphen mit Baumeigenschaften
seit 1995	Wissenschaftlicher Assistent (C1) am Institut für Theoretische Informatik an der Universität Rostock
1997	Einmonatiger Forschungsaufenthalt am Wissenschaftlichen Zentrum der IBM in Heidelberg *Thema:* Beschreibung eines Vehicle Routing Problems

Erklärungen gemäß §3, Absatz 1, Punkt 7 und 8 der Promotionsordnung

1. Hiermit erkläre ich, daß ich die eingereichte Dissertation selbständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

2. Hiermit erkläre ich, daß ich mich weder an der Universität Rostock noch an einer anderen Universität um den Doktorgrad beworben habe.

Rostock, 07.12.2001