On Equidomination in Graphs

Inaugural-Dissertation

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Fabian Johannes Senger

aus Damme

Berichterstatter:	Prof. Dr. Rainer Schrader Prof. Dr. Oliver Schaudt

Tag der mündlichen Prüfung: 19.12.2017

Abstract

A graph $G = (V, E)$ is called equidominating if there exists a value $t \in \mathbb{N}$ and a weight function $\omega\colon V \to \mathbb{N}$ such that the total weight of a subset $D \subseteq V$ is equal to t if and only if D is a minimal dominating set. Further, ω is called an equidominating function, t a target value and the pair (ω, t) an equidominating structure. To decide whether a given graph is equidominating is referred to as the EQUIDOMINATION problem.

First, we examine several results on standard graph classes and operations with respect to equidomination. Furthermore, we characterize hereditarily equidominating graphs. These are the graphs whose every induced subgraph is equidominating. For those graphs, we give a finite forbidden induced subgraph characterization and a structural decomposition. Using this decomposition, we state a polynomial time algorithm that recognizes hereditarily equidominating graphs.

We introduce two parameterized versions of the EQUIDOMINATION problem: the k-EQUIDOMINATION problem and the TARGET-t EQUIDOMINATION problem. For $k \in \mathbb{N}$, a graph is called k-equidominating if we can identify the minimal dominating sets using only weights from 1 to k. In other words, if an equidominating function with co-domain $\{1, \dots, k\}$ exists. For $t \in \mathbb{N}$, a graph is said to be target-t equidominating if there is an equidominating structure with target value t.

For both parameterized problems we prove fixed-parameter tractability. The first step for this is to achieve the so-called pseudo class partition, which coarsens the twin partition. It is founded on the requirement that vertices from different blocks of the partition cannot have equal weights in any equidominating structure. Based on the pseudo class partition, we state an *XP* algorithm for the parameterized versions of the EQUIDOMINATION problem.

The second step is the examination of three reduction rules – each of them concerning a specific type of block of the pseudo class partition – which we use to construct problem kernels. The sizes of the kernels are bounded by a function depending only on the respective parameter. By applying the *XP* algorithm to the kernels, we achieve *FPT* algorithms.

Finally, we generalize the property k-equidominating by allowing k different weights not larger than a positive integer $W \in \mathbb{N}$ and show analogous results for the so-called k-W-EQUIDOMINATION problem based on this generalization.

The concept of equidomination was introduced nearly 40 years ago, but hardly any investigations exist. With this thesis, we want to fill that gap. We may lay the foundation for further research on equidomination.

Kurzzusammenfassung

Ein Graph $G = (V, E)$ heißt äquidominierend, wenn es eine Zahl $t \in \mathbb{N}$ und eine Gewichtsfunktion $\omega\colon V \to \mathbb{N}$ gibt, so dass das Gesamtgewicht einer Teilmenge $D \subseteq V$ genau dann gleich t ist, wenn D eine minimal dominierende Menge ist. In diesem Fall wird ω als eine äquidominierende Funktion bezeichnet, t als ein Zielwert und das Paar (ω, t) als eine äquidominierende Struktur. Das Äquidominanz Problem bezeichnet das Problem zu entscheiden, ob ein gegebener Graph äquidominierend ist oder nicht.

Zunächst arbeiten wir einige Ergebnisse bezüglich Standardgraphen und verschiedener Operationen im Kontext von Äquidominierung aus. Weiterhin charakterisieren wir vererbende äquidominierende Graphen. Das sind die Graphen, deren induzierte Teilgraphen auch äquidominierend sind. Für diese Graphen geben wir eine Charakterisierung anhand von endlich vielen verbotenen induzierten Subgraphen an und eine strukturelle Zerlegung. Auf dieser Zerlegung basiert ein schneller Algorithmus für die Identifizierung von vererbenden äquidominierenden Graphen.

Wir führen zwei parametrisierte Versionen des Äquidominanz Problems ein: das k-Äquidominanz Problem und das Zielwert-t Äquidominanz Problem. Für $k \in \mathbb{N}$ wird ein Graph k-äquidominierend genannt, wenn wir die minimal dominierenden Mengen mit Gewichten von 1 bis k identifizieren können. Mit anderen Worten, wenn eine äquidominierende Funktion mit Zielmenge $\{1, \ldots, k\}$ existiert. Für $t \in \mathbb{N}$ heißt ein Graph Zielwert-t äquidominierend, wenn es eine äquidominierende Struktur mit Zielwert t gibt.

Wir zeigen, dass beide parametrisierte Probleme in der Komplexitätsklasse *FPT* liegen. Der erste Schritt hierfür ist die Erforschung der sogenannten Pseudo-Klassen-Partition. Diese Partition ist eine Vergröberung der Partition in Zwillinge. Die Pseudo-Klassen-Partition begründet sich auf der Forderung, dass Knoten aus verschiedenen Blöcken der Partition in keiner äquidominierenden Struktur die gleichen Gewichte haben können. Basierend auf der Pseudo-Klassen-Partition leiten wir einen *XP*-Algorithmus für die parametrisierten Versionen des Äquidominanz Problems her.

Der zweite Schritt besteht in der Ausarbeitung dreier Reduktionsregeln, mit denen wir Problemkerne konstruieren. Die Reduktionsregeln behandeln jeweils einen bestimmten Blocktyp der Pseudo-Klassen-Partition. Die Größen der Kerne sind durch eine Funktion begrenzt, die nur vom jeweiligen Parameter abhängt. Durch Anwendung des *XP*-Algorithmus auf die Problemkerne erhalten wir *FPT*-Algorithmen.

Des Weiteren verallgemeinern wir k-Äquidominierung, indem wir k verschiedene Gewichte zulassen, die nicht größer als eine natürliche Zahl $W \in \mathbb{N}$ sein dürfen. Wir erzielen analoge Ergebnisse für das daraus resultierende k-W-ÄQUIDOMINANZ Problem.

Obwohl Äquidominierung vor fast 40 Jahren eingeführt wurde, gibt es bisher kaum Ergebnisse dazu. Mit dieser Arbeit wollen wir diese Lücke füllen. Möglicherweise legen wir damit den Grundstein für weitere Untersuchungen zu Äquidominierung.

Contents

Chapter 1

Introduction

The first appearance of domination in graphs goes back to the 1850's, even though it was not considered as domination in these days. Chess players posed the question of how many queens are needed such that every square of the 8×8 chess field is occupied by a queen or reachable by a queen within one move. On a chessboard, a queen can move to every square in horizontal, vertical or diagonal direction.

The embedding of this question into the framework of graph theory leads to the concept of domination. Firstly, this embedding and the mathematical formalization of domination was done by Berge [4] and Ore [54] in 1962. While Ore used the term domination right from the beginning, Berge spoke of external stability at that time (following internal stability, which he used for so-called stable sets). However, it was domination that prevailed.

Let us follow the ideas of Berge and Ore and formulate the question in graph-theoretic language. For that, let every square of the chessboard be a vertex. We connect two vertices by an edge if it is possible to get from one square to the other within a queen move. The obtained graph is called the queen-graph (and can be defined analogously for every chess piece).

With this graph-theoretical framework, the question of the chess players can be reformulated as follows:

> What is the minimum size of a subset of vertices such that every vertex of the queen-graph is an element of the subset or is connected to a vertex of the subset?

By this question, we have just stated the definition of a dominating set. A subset D is a dominating set if every vertex not in D is connected – or adjacent, as graph-theorists call it – to a vertex of D.

A dominating set of minimum size is called a minimum dominating set, while a dominating set that does not contain another dominating set is called a minimal dominating set. This means that we cannot delete a vertex of a minimal dominating set without losing the property of being dominating. However, a minimal dominating set can be of significantly larger size than a minimum dominating set.

Besides chess, lots of other applications for domination exist (see [35]). For example, let the vertices of a graph represent persons such that an edge between two persons exists if they know each other. Such a graph is often called social graph [7]. We want to determine a committee such that every person not in the committee knows a person in it. Again, such a committee corresponds to a dominating set. Further, we could also want the committee to have as little members as possible. Then we are looking for a minimum dominating set in the graph.

Another example is the distribution of facilities (such as police or fire departments, schools, supermarkets, radio stations and so on) to demand locations (such as districts, villages, ...). For every location, there must be a facility nearby. In this context, "nearby" means at the location itself or at a neighboring location. In general, computing the minimum number of needed facilities can get quite hard. Given that circumstance, it is a reasonable goal to at least avoid unnecessary facilities, which means we are indeed looking for an (inclusion-wise) minimal dominating set.

While the main stream of the research on dominating sets in graphs focuses on the optimization aspects of the problem, there are several interesting graph classes defined around this concept. For example, the classes of efficient dominating graphs [11], of well-dominated graphs [27], of domination perfect graphs [61], of upper domination perfect graphs [32] and of strong domination perfect graphs [58].

Another example is the class of domishold graphs, introduced 1978 by Benzaken and Hammer in [3]. These are the graphs for which there are positive weights associated to the vertices of the graph such that a subset D of vertices is dominating if and only if the sum of the weights of the vertices of D (also called the total weight of D) exceeds a certain threshold t. In other words, the characteristic vectors of the dominating sets are exactly the zero-one solutions of a linear inequality, where the coefficients of the inequality correspond to the weights of the vertices and the right-hand side to the threshold (see [13], [15], [44] and [46] for more details).

Motivated by this concept, Payan asked in 1980 whether there are graphs for which the characteristic vectors of the minimal dominating sets are the zero-one solutions of a linear equality [56]. Equivalently, we are looking for a weight function such that not only every minimal dominating set has a specific total weight. We further require that every subset of vertices with that specific total weight is a minimal dominating set. Such graphs Payan named equidominating.

Equidominating graphs are the principal topic of this thesis. As mentioned before, the concept was introduced nearly forty years ago. However, to our knowledge, there are only a few results on equidomination. Payan stated in the introducing paper [56] a characterization of the class of graphs that are domishold and equidominating. Further, he showed that threshold graphs are equidominating. There is no direct relation between domishold and equidominating. There are equidominating graphs that are not domishold and vice versa [44]. Moreover, the complexity of deciding whether a given graph is equidominating is (so far) unknown.

One advantage of having an equidominating structure – that is a weight function and a specific total weight of exactly the minimal dominating sets – of a graph at hand is that one can check whether a given vertex subset is a minimal dominating set in linear time. One simply has to consider the sum of the weights of the vertices of the subset. It can be worthwhile to examine an equidominating structure of a graph if one has to check subsets for being minimal dominating over and over again, for example as a subroutine in some algorithm.

In the course of this thesis, we explore several issues regarding equidominating. On the one hand, these issues concern general results with respect to standard graph classes and operations, and the characterization of hereditarily equidominating graphs. On the other hand, as the main topic of this thesis, we embed equidomination into (parameterized) complexity theory.

1.1 Relation to Equistability

A stable set is a set of pairwise non-adjacent vertices and a maximal stable set is not contained in another stable set. In [56], Payan also brought up the term equistable. This graph property is defined analogously to equidominating with respect to maximal stable sets. That is, a graph is equistable if there is a weight function such that maximal stable sets are identified by their total weights. Since every maximal stable set is dominating, maximal stable sets are also called independent dominating sets (in the field of graph theory independent is used synonym to stable). This already indicates the correlation between domination and stability in general and hence also between equidomination and equistability.

In contrast to equidomination, there are quite a few investigations on equistability. The first paper on equistability, that followed the introducing paper, appeared in 1994 by Mahadev, Peled and Sun [45]. There, the authors gave necessary and sufficient conditions for equistability, characterized the equistability of various graph classes and introduced the stronger property strongly equistable. They conjectured that the class of equistable graphs not only contains the class of strongly equistable graphs but that both classes are identical.

In 2009, Orlin extended this conjecture (see [48]). He proved that every so-called general partition graph is equistable and conjectured that the converse is also true. However, both conjectures were recently disproved by Milanič and Trotignon [51]. Various other papers were published in this context, see for example [8] and [42].

Another scope of the research on equistability treats complexity issues (see [36], [43] and [49]). We only want to mention here that (so far) the complexity of the recognition of equistable is unknown. The interested reader is referred to Chapter 4, where we also consider this topic.

Several other papers deal with the characterization of equistability of different graph classes, such as chordal graphs [57], distance-hereditary graphs [38], EPT graphs [1] or series-parallel graphs [37].

Even though there are many analogies between equistability and equidomination, on the other hand there are significant differences in the case of equidomination. It turns out that the concept of equidomination case bears its own substantial difficulties which we overcome with new methods.

1.2 Outline

This thesis is organized as follows. After this introducing chapter, we state basic graph-theoretic definitions and notations as well as the needed essentials of complexity theory in Chapter 2.

Chapter 3 begins with an introduction to the concept equidomination, starting by its first appearance in 1980 and stating the few, yet existing results. Furthermore, we show that no induced subgraph is forbidden for equidominating graphs. Then, we achieve several results with respect to equidomination: on the one hand, these results regard standard graph classes and the question whether they are equidominating or not. On the other hand, we examine numerous operations on graphs – like adding vertices and joining graphs – and to what extent they are compatible with equidomination.

Next, we turn our attention to the class of hereditarily equidominating graphs. In these graphs every induced subgraph is equidominating. We obtain a characterization of this class in terms of forbidden induced subgraphs as well as a structural decomposition. Using this decomposition, we state a polynomial time recognition algorithm for hereditarily equidominating graphs. Chapter 3 ends with a proof of the existence of a linear time recognition algorithm.

In Chapter 4, we embed equidomination into the framework of complexity theory. We define the EQUIDOMINATION problem – decide whether a given graph is equidominating or not – and discuss some related topics. Unfortunately, the computational complexity of this problem is unknown. This is why we specify the property equidominating in two ways: on the one hand, we allow only weights that are not larger than a specific value. And on the other hand, we prescribe the specific total weight every minimal dominating set must have. These two specifications naturally lead to two parameterized versions of the EQUIDOMINATION problem: the k-EQUIDOMINATION problem and the TARGET-t EQUIDOMINATION problem.

In order to achieve complexity results for the parameterized problems, we want to find a partition of the vertices of a graph such that vertices of different blocks of the partition cannot have the same weight with respect to any equidominating function. We obtain the so-called pseudo class partition. The pseudo class partition is a coarsening of the twin partition. Its examination is one of the primary outcomes of our research and one of the centerpieces to prove fixed-parameter tractability of the parameterized problems. At the end of Chapter 4, we use the pseudo class partition to state an *XP* algorithm which solves the k-EQUIDOMINATION problem. The *XP* algorithm can also be applied to the TARGET-t EQUIDOMINATION problem.

In Chapter 5, we examine three reduction rules, the second centerpiece to show fixed-parameter tractability. With these rules, we can reduce the size of a given graph without changing its property of being k-equidominating and target-t equidominating. Each rule deals with a specific type of block of the pseudo class partition.

As the main results of this thesis, we deduce complexity results for the two parameterized problems k-EQUIDOMINATION and TARGET-t EQUIDOMINATION. We show that both problems are fixed-parameter tractable. By applying the three reduction rules, we construct problem kernels for both parameterized versions of the EQUIDOMINATION problem. Moreover, using the *XP* algorithm presented in Chapter 4, we obtain *FPT* algorithms.

Chapter 5 ends with a generalization of k-equidomination. We allow k different weights that do not necessarily need to be the numbers $1, \ldots, k$ but are bounded by some integer $W > k$. We show fixed-parameter tractability for the resulting problem k-W-EQUIDOMINATION, too.

The thesis closes with a summary of our results and a detailed as well as a wider outlook to possible future research in Chapter 6.

Chapter 2

Preliminaries

In Section 2.1, we give the basic graph-theoretic definitions and notations that we use in this thesis. Section 2.2 is devoted to complexity theory, where we first give a brief introduction to the fundamentals and afterward to fixed-parameter tractability. An index is provided at the end of the thesis such that the reader is able to find every definition again.

2.1 Basic Definitions and Notations

Even though most of the following notations are standard, we want to state in this section everything which is needed to understand this thesis. If a (rather specific) definition is just needed once in this thesis, we only mention it at the passage where we use it.

We begin by stating general mathematical notations that do not necessarily depend on graphs. The cardinality of a set M is denoted by $|\boldsymbol{M}|$. If we want to emphasize that a subset M' of M is a proper subset, we write $\boldsymbol{M' \subset M}$ and that we unite two disjoint sets M_1 and M_2, we write $\boldsymbol{M_1 \dot\cup M_2}$ (instead of $M' \subseteq M$ and $M_1 \cup M_2$). We use the abbreviation $[\boldsymbol{k}] := \{1, \ldots, k\}$ for a natural number $k \in \mathbb{N}$.

A **partition** of a set M is a family $\mathcal{P} = \{P_1, \ldots, P_s\}$ of subsets of M such that $\bigcup_{i=1}^{s} P_i = M$ and $P_i \cap P_j = \emptyset$ for i, $j \in [s]$ with $i \neq j$. The elements of $\mathcal{P}$ are called **blocks**. The equivalence classes of an equivalence relation on M form a partition. In this case, we call the blocks also **classes** of the partition. Let $\mathcal{P}$ and $\mathcal{P}'$ be two partitions of a set M such that two elements x, $y \in M$ lie in the same block of $\mathcal{P}$ if x and y lie in the same block of $\mathcal{P}'$. Then, we call $\mathcal{P}$ a **coarsening** of $\mathcal{P}'$ and $\mathcal{P}'$ a **refinement** of $\mathcal{P}$. Furthermore, let $(m_1, \ldots, m_n)$ be an ordering of the set M. We call a vector $\chi \in \{0, 1\}^n$ a **characteristic vector** of M. Each characteristic vector determines a subset M' of M and vice versa: $m_i \in M'$ if and only if the i-th component χ_i of χ equals 1.

A **weight function** on a set M is a function $\omega\colon M \to \mathbb{N}$. For $m \in M$ we call $\omega(m)$ the **weight of m**. For a subset $M' \subseteq M$ we define $\boldsymbol{\omega(M')} := \sum_{m \in M'} \omega(m)$. For a better distinction, we call $\omega(M')$ the **total weight of M'** if M' is a subset.

Now, we come to graph-theoretic definitions and notations. For a complete and more detailed introduction to graph theory, the interested reader is referred to the books of Brandstädt et al. [10], Diestel [23] and Korty and Vygen [39].

A **graph** is a tupel $G = (V, E)$ with V being a non-empty set and $E \subseteq V \times V$ (with $\times$ denoting the Cartesian product). The elements of V are called **vertices** and the elements of E **edges**. Further, we call V the **vertex set** and E the **edge set**. As we only work with undirected graphs, we consider an edge as an unordered pair and write $e = vw$ for an edge $e \in E$ and two vertices $v, w \in V$. We also refer to the set of vertices of G as $\boldsymbol{V(G)}$ and to the set of edges of G as $\boldsymbol{E(G)}$. All graphs considered in this thesis are finite, and without any loops or parallel edges. This means that V is finite, $vv \notin E$ for all $v \in V$ and $e \neq f$ for all $e, f \in E$, respectively. For two vertices $v, w \in V$ we say that v and w are **adjacent** if $vw \in E$ and that v and w are **incident** to the edge vw.

In the remainder of this section, let $G = (V, E)$ be a graph. The **complement** of G is the graph with vertex set V and edge set $(V \times V) \setminus E$ and is denoted by $\boldsymbol{\overline{G} = (V, \overline{E})}$. A graph H is said to be **isomorphic** to G if there is a bijection $\phi\colon V(G) \to V(H)$ with $vw \in E(G)$ if and only if $\phi(v)\phi(w) \in E(H)$. If G and H are isomorphic we write $\boldsymbol{G \cong H}$. In general, we do not distinguish between isomorphic graphs.

For a subset $V' \subseteq V$, the (vertex) **induced subgraph of $\boldsymbol{G}$** is denoted by $\boldsymbol{G[V']}$ and defined as the graph with vertex set V and edge set $\{vw \in E \mid v, w \in V'\}$. A graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subset E(G[V'])$ is simply called **subgraph of $\boldsymbol{G}$**. A graph induced by a subset of edges is the graph induced by the set of vertices that are incident to an edge of the subset.

Next, we will define several abbreviations. For that, let $v \in V$, $e \in E$, $V' \subseteq V$ and $E' \subseteq E$. By $\boldsymbol{G - v}$ and $\boldsymbol{G - V'}$ we denote the induced subgraphs $G[V \setminus \{v\}]$ and $G[V \setminus V']$, respectively. By $\boldsymbol{G - e}$ and $\boldsymbol{G - E}$ we denote the subgraphs $(V, E \setminus \{e\})$ and $(V, E \setminus E)$, respectively. Furthermore, for $E'' \subseteq V \times V$, $\boldsymbol{G + E''}$ denotes the graph $(V, E \cup E'')$. If $x \in V'$ and $y \notin V'$, then we write $\boldsymbol{V' - x + y}$ for $(V' \setminus \{x\}) \cup \{y\}$.

A **complete graph** is a graph where every possible edge exists (that is $E = V \times V$). We denote the complete graph on $n \in \mathbb{N}$ vertices by $\boldsymbol{K_n}$. The complement $\boldsymbol{\overline{K_n}}$ of a complete graph is called **edgeless graph**. A graph is called **bipartite** if there is a partition of the vertices into two blocks such that each block induces an edgeless graph. We call the two blocks **color classes**. If every edge between the vertices of the color classes exists, then the graph is said to be **complete bipartite**. A complete bipartite graph with color classes of size m and n ($m, n \in \mathbb{N}$) is denoted by $\boldsymbol{K_{m,n}}$. The graphs $K_{1,n}$ are also called **star graphs**. Let $V_n = (v_1, \ldots, v_n)$. The graph with vertex set V_n and edge set $\{v_i v_{i+1} \mid i = 1, \ldots, n-1\}$ is the **path $\boldsymbol{P_n}$**. The **cycle $\boldsymbol{C_n}$** is the graph $P_n + v_1 v_n$. Let $V_{2n} = (v_1, v_1', \ldots, v_n, v_n')$. By $\boldsymbol{T(2n, n)}$ we denote the graph with vertex set V_{2n} and edge set $(V_{2n} \times V_{2n}) \setminus \{v_i v_i' \mid i = 1, \ldots, n\}$. This is a particular **Turàn graph** (see [23] for a complete definition of Turàn graphs). The graph $T(2n, n)$ is also called the cocktail party graph: every vertex is a guest of a party with n couples and the edges symbolize the handshakes between all guests, where partners do not shake hands.

For a vertex $v \in V$, we call $\boldsymbol{N(v)} := \{w \in V \mid vw \in E\}$ the **open neighborhood** of v. Every vertex of $N(v)$ is said to be a **neighbor** of v. The **closed neighborhood** of v is defined as $\boldsymbol{N[v]} := N(v) \cup \{v\}$. We also use these terms for subsets $S \subseteq V$ of vertices, then $\boldsymbol{N(S)} := \bigcup_{v \in S} N(v)$ and $\boldsymbol{N[S]} := N(S) \cup S$. Further, we define the **private neighbor set** of $v \in S$ as $\boldsymbol{pn[v, S]} := N[v] \setminus N[S \setminus \{v\}]$ and every element of $pn[v, S]$ is called a **private neighbor** of v.

A subset S of the vertices of a graph is called a **dominating set** or simply **dominating**, if every vertex of the graph is an element of S or adjacent to a vertex of S. Since dominating sets are essential, we give some equivalent definitions. A subset $S \subseteq V$ is a dominating set if and only if:

(i) $N[S] = V$,

(ii) $|N(v) \cap S| \geq 1$ for every vertex $v \in V \setminus S$,

(iii) $|N[v] \cap S| \geq 1$ for every vertex $v \in V$.

If a dominating set D does not properly contain another dominating set, it is called a **minimal dominating set**. That is, each $D' \subset D$ is not a dominating set. Further, D is a minimal dominating set if and only if $N[S] = V$ and $pn[v, D] \neq \emptyset$ holds for all $v \in D$. We want to emphasize that is the rest of the thesis we work with minimal dominating sets and not with minimum dominating sets (which is a dominating set of minimal cardinality). We say that a vertex $v \in V$ **dominates** itselft and all its neighbors. Also, every subset of vertices $S \subseteq V$ dominates each vertex of $N[S]$.

A **stable set** is a subset of pairwise non-adjacent vertices and a **clique** is a subset of pairwise adjacent vertices. In other words, a stable set induces an edgeless graph and a clique induces a complete graph. Analogously to a minimal dominating set, a stable set is said to be a **maximal stable set** if it is not properly contained in another stable set. It is easy to see that every maximal stable set is a minimal dominating set.

Let v, $w \in V$ be two vertices. The **degree** of v is defined as $\mathbf{deg}(\boldsymbol{v}) := |N(v)|$. The length of a shortest path between v and w is denoted by $\mathbf{dist}(\boldsymbol{v}, \boldsymbol{w})$ and called the **distance** of v and w. We set $\text{dist}(v, w) := \infty$ if no such path exists. The **diameter** of a graph is the longest shortest path, that is $\max_{v,w \in V}(\text{dist}(v, w))$. If $\deg(v) = |V| - 1$ or $\deg(v) = 0$, then v is said to be a **universal** or **isolated** vertex, respectively. We call v a **pendant** vertex (of w), if $\deg(v) = 1$ (and $vw \in E$). If $\text{dist}(x, y) < \infty$ for all x, $y \in V$, then G is called **connected**. We also call a subset $S \subseteq V$ of vertices connected if $\text{dist}(x, y) < \infty$ for all x, $y \in S$. Further, a maximal induced subgraph of G that is connected is called a **component** of G.

Now, let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. The **disjoint union** $\boldsymbol{G_1 \cup G_2}$ of G_1 and G_2 is the graph $(V_1 \cup V_2, E_1 \cup E_2)$. The **complete union** $\boldsymbol{G_1 + G_2}$ of G_1 and G_2 is the graph $(V_1 \cup V_2, E_1 \cup E_2 \cup E_{1,2})$ with $E_{1,2} = \{v_1 v_2 \mid v_1 \in V_1, v_2 \in V_2\}$. A **chain graph** is a bipartite graph where the neighborhoods of the vertices of either color class are comparable with respect to inclusion. This means that orderings of the vertices of the color classes $\{v_1, \ldots, v_n\}$ and $\{w_1, \ldots, w_m\}$ of a chain graph exist such that $N(v_1) \subseteq \ldots \subseteq N(v_n)$ and $N(w_1) \subseteq \ldots \subseteq N(v_m)$. Let U_i be the (possibly empty) set of universal vertices of G_i, for $i \in \{1, 2\}$. Let B be any chain graph with color classes U_1 and U_2. We call the graph $(G_1 \cup G_2) + E(B)$ a **chain-join** of G_1 and G_2. Note that the disjoint union of any two graphs is a particular chain-join of these two graphs. Let G be a chain-join of G_1 and G_2. We remark that even though we use a chain graph to join G_1 and G_2 to obtain G, the induced subgraph $G[U_1 \cup U_2]$ is not a chain graph since the former color classes of B are cliques in G. A **co-chain graph** is the complement of a chain graph.

A subset of vertices $V' \subseteq V$ is said to be a **module** if $N(v) \setminus V' = N(w) \setminus V'$ for all $v, w \in V'$. Clearly $\emptyset$, V and the sets $\{v\}$ for all $v \in V$ are modules. Those are the **trivial** modules. A graph is called **prime** if each of its modules is trivial. The **modular decomposition** of a graph is a decomposition of the vertex set into modules. This decomposition can be done recursively, leading to the so-called decomposition tree, which represents all modules of a graph. We only need a special kind of modules and a deeper examination of this topic goes beyond the scope of this thesis. Actually, modular decomposition is not restricted to graphs but applicable to numerous discrete structures. The interested reader is referred to the book of Brandtstädt et al. [10] for a brief introduction and to the survey papers of Habib and Paul [33] and Möhring and Radermacher [52] for a deeper insight.

If necessary, we add a subscript to any term to clarify in which graph we consider term (for example, we write $N_G(v)$ to point out that we consider the neighborhood of v in G).

2.2 Complexity Theory

In this section, we give a brief introduction to the wide field of complexity theory. We state formal definitions as well as rather informal interpretations, and how we make use of the described objects for our purposes. Furthermore, we introduce the concept of fixed-parameter tractability by citing the elemental definitions and a basic result that we use to prove our main results.

2.2.1 Fundamentals

Let Σ be a finite set. We call Σ an **alphabet**. By Σ^* we denote the set of strings of every possible length whose symbols are elements of Σ, that is $\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$. Now, a **language $\boldsymbol{X}$ over $\boldsymbol{\Sigma}$** is a subset of Σ^*. Here, we are not interested in technical issues how Σ exactly looks like or which language underlies the considered decision problems. For our purposes it is enough to think of Σ as $\{0, 1\}$. Then, Σ^* is the set of all binary strings. With the set of all binary strings we can represent and encode everything we work with.

Definition 2.2.1 ([39], Definition 15.7)**.** *A* ***decision problem*** *is a pair $\mathcal{P} = (X, Y)$, where X is a language decidable in polynomial time and $Y \subseteq X$. The elements of X are called* ***instances*** *of $\mathcal{P}$; the elements of Y are* ***yes-instances****, those of $X \setminus Y$ are* ***no-instances****.*

An ***algorithm*** *for a decision problem (X, Y) is an algorithm computing the function $f\colon X \to \{0, 1\}$, defined by $f(x) = 1$ for $x \in Y$ and $f(x) = 0$ for $x \in X \setminus Y$.*

Here, decidable in polynomial time means that for all $x \in \Sigma^*$ it can be decided in polynomial time (see below) if $x \in X$.

For a better understanding, we define the following decision problem and relate it to Definition 2.2.1.

Dominating Set:
Instance: A graph G and $k \in \mathbb{N}$.
Problem: Decide whether G has a dominating set of cardinality k.

As mentioned above we are not interested in how the underlying language X of Dominating Set exactly looks like. For us it is only important that X contains all pairs (G, k), where G is a graph and $k \in \mathbb{N}$, and that Y is the set of all graphs with a dominating set of cardinality k. One algorithm for Dominating Set is the brute force approach: compute all subsets of vertices of cardinality k of a graph and check if one of them is a dominating set. If so, return 1 (or YES), otherwise, return 0 (or NO).

The central aspect of complexity theory is to determine how well a decision problem can be solved and which decision problems can be solved efficiently. But what does efficiently mean? For that we need the **size** of an instance of a decision problem. Formally, the size of an element $x \in X$ is the length of the string x. For us, if the instance is a graph, then its size is the number of vertices and edges. We remark again that we are not concerned how to encode a graph (for an interesting comment on this topic see Example 8.5 of [55]).

Now, a decision problem is said to be **polynomial time solvable** if there is an algorithm whose running time is bounded by a polynomial in the instance/input size. **Efficiently** is used synonymously with polynomial time solvable. The class of decision problems that are polynomial time solvable is denoted by $\boldsymbol{P}$.

To measure the **running time** of an algorithm precisely we use the so-called Big-Oh notation. For that we define for a function $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ the set

$$\mathcal{O}(f) \coloneqq \{g\colon \mathbb{N}_0 \to \mathbb{N}_0 \mid \exists c, n_0 \in \mathbb{N}\ \forall n \geq n_0\colon g(n) \leq cf(n)\}\ .$$

This means that $\mathcal{O}(f)$ contains all functions that are asymptotically bounded by (a multiple of) f. Note that we do not measure the actual time an algorithm is running since this is highly depending on the used hardware. Instead we are counting the number of so-called elementary steps (arithmetic operations like addition/multiplication, comparison, branching instructions, ...). Slightly abusing notation, it is common to write $f = \mathcal{O}(g)$ and $f \leq \mathcal{O}(g)$ instead of $f \in \mathcal{O}(g)$. We also say that f is of **order** g.

For the sake of completeness we at least want to mention that the extensive field of machine models underlies the analysis of running times of algorithms and thus the classification of decision problems. However, as we do not need the technical details, it is omitted here. The reader interested in this topic is referred to the survey by Boas [5].

Is it possible to find a polynomial time algorithm for every decision problem? To quote Papadimitriou and Steiglitz [55]:

> "The time has now come to meet the most prominent failures of this approach; problems, that is, for which no efficient algorithm is known. In doing so, we shall develop a beautiful theory that unifies these failures into deep mathematical conjecture."

If we take a look at the above-given algorithm for DOMINATING SET, it is easy to see that this algorithm does not have a polynomial running time. For a graph on n vertices, we have to analyze every subset of vertices of size k, of which $\binom{n}{k} = \mathcal{O}(n^k)$ many exists (note that k is part of the instance). DOMINATING SET is contained in another complexity class, which is defined as follows. The class of decision problems for which we can verify in polynomial time that a yes-instance is indeed a yes-instance is denoted by ***NP***. The verification is done with a so-called **certificate**.

For DOMINATING SET such a certificate is simply a dominating set of size k of the considered graph. Given such a set, we have to check if it is of size k and if it is a dominating set. If so, the graph is a yes-instance. Since checking both conditions can be done in polynomial time, DOMINATING SET is in *NP*.

If we can transform a decision problem $\mathcal{P}_1$ to a decision problem $\mathcal{P}_2$ in polynomial time such that yes-instances of $\mathcal{P}_1$ are transformed to yes-instances of $\mathcal{P}_2$, and no-instances of $\mathcal{P}_1$ are transformed to no-instances of $\mathcal{P}_2$, then we said that $\mathcal{P}_1$ **reduces/is reducible to** $\mathcal{P}_2$. In this case, if there is a polynomial time algorithm for $\mathcal{P}_2$, then there is also one for $\mathcal{P}_1$: we transform a given instance of $\mathcal{P}_1$ to an instance of $\mathcal{P}_2$ and solve it with the algorithm for $\mathcal{P}_2$, both in polynomial time.

A decision problem $\mathcal{P}$ is called ***NP*-hard**, if every problem of *NP* is reducible to $\mathcal{P}$. If in addition $\mathcal{P} \in NP$, then $\mathcal{P}$ is said to be ***NP*-complete**.

Following the definitions, we get $P \subseteq NP$. The question whether there are problems in *NP* that are not in *P* or whether $P = NP$ is the most important issue in complexity theory. The question is open for decades and most people believe that $P \neq NP$. This belief is the "deep mathematical conjecture" Papadimitriou and Steiglitz speak about. The DOMINATION SET problem is *NP*-complete (see Theorem 1.7 in [35]), so it is most likely that there exists no polynomial time algorithm.

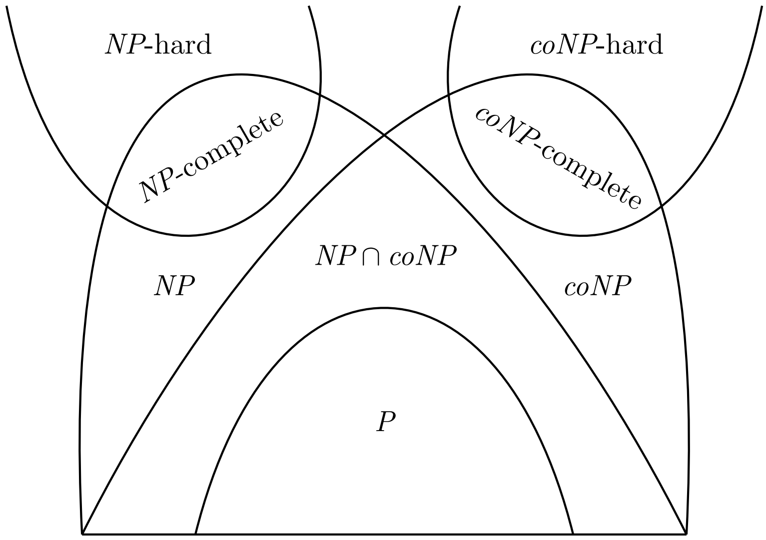


Figure 2.1: Relation (most likely) between the complexity classes *P*, *NP*, *NP*-complete, *NP*-hard, *coNP*, *coNP*-complete and *coNP*-hard

We close this subsection with the introduction of another complexity class. The class of decision problem for which we can verify no-instances in polynomial time is denoted by ***coNP***. Analogously, a decision problem is called ***coNP*-hard**, if every problem of *coNP* is reducible to it and ***coNP*-complete** if it is additionally in *coNP*.

For an overview over the introduced complexity classes see Figure 2.1 [39]. The reader interested in the topic of complexity theory is referred to the book [31] for a deeper insight.

2.2.2 Parameterized Complexity

The concept of parameterized complexity is used to refine the analysis of problems that are *coNP*-hard. It was formally introduced by Downey and Fellows in the 1990s [24]. Parameterized complexity is motivated by the fact that often a large part of the running time of an algorithm depends on a problem-specific parameter and that those parameters often appear small in practical applications.

We follow the notation of Flum and Grohe [28]. First, we need the definition of a parameterized problem. Note that Flum and Grohe use a slightly different notation of decision problems as Vygen and Korte, who we cited in the previous subsection (in the following, (X, Y) of Definintion 2.2.1 corresponds to (Σ^*, Q)).

Definition 2.2.2 ([28], Definition 1.1)**.** *Let* Σ *be a finite alphabet.*

(i) A ***parameterization*** *of* Σ^* *is a mapping* $\kappa\colon \Sigma^* \to \mathbb{N}$ *that is polynomial time computable.*

(ii) A ***parameterized problem*** *(over* Σ*) is a pair* (Q, κ) *consisting of a set* $Q \subseteq \Sigma^*$ *of strings over* Σ *and a parameterization* κ *of* Σ^**.*

Several parameterizations are possible for a decision problem. For example, let (G, k) be an instance of Dominating Set. Then one possible parameterization is $\kappa((G, k)) = k$, which leads to the following parameterized problem (compare [28]):

p-Dominating Set:

Instance:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide whether G has a dominating set of cardinality k.

The next two definitions introduce two complexity classes for parameterized problems. The crucial thing here is to separate the size of the parameter in the running time from the size of the rest of the problem instance.

Definition 2.2.3 ([28], Definition 2.22)**.** *Let* (Q, κ) *be a parameterized problem over the alphabet* Σ*. Then* (Q, κ) *belongs to the class* ***XP*** *if there is a computable function* $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ *and an algorithm that, given* $x \in X^*$*, decides if* $x \in Q$ *in at most*

$$|x|^{f(\kappa(x))} + f(\kappa(x))$$

steps.

We call an such an algorithm an ***XP*** **algorithm**.

Definition 2.2.4 ([28], Definition 1.4)**.** *Let* Σ *be a finite alphabet and* $\kappa\colon \Sigma^* \to \mathbb{N}$ *a parameterization.*

(i) An algorithm $\mathbb{A}$ *with input alphabet* Σ *is an* ***FPT algorithm with respect to*** $\boldsymbol{\kappa}$ *if there is a computable function* $f\colon \mathbb{N} \to \mathbb{N}$ *and a polynomial* $p \in \mathbb{N}_0[X]$ *such that for every* $x \in \Sigma^*$*, the running time of* $\mathbb{A}$ *on input* x *is at most*

$$f(\kappa(x)) \cdot p(|x|)\ .$$

(ii) A parameterized problem (Q, κ) *is* ***fixed-parameter tractable*** *if there is an FPT algorithm with respect to* κ *that decides* Q*.*

FPT *denotes the class of all fixed-parameter tractable problems.*

The function f is typically super-polynomial and even functions like $f(k) = 2^{2^{2^{2^k}}}$ are legitimate.

The difference between Definition 2.2.3 and Definition 2.2.4 is that for an FPT algorithm we can separate the size of the input completely from the parameter into two factors. In contrast, for an *XP* algorithm it is only demanded that the size of the input does not appear as an exponent in the running time. For example, if n is the input size and k the parameter, then an algorithm with running time n^k is an *XP* algorithm and one of runnig time $2^k \cdot n$ an FPT algorithm. It holds that $FPT \subseteq XP$ (see Proposition 3.2 in [28]).

It follows that p-Dominating Set is in *XP*. But it is most likely, that p-Dominating Set is not in FPT (unless $P = NP$). However, if we choose another parameterization (one that includes the maximum degree of a graph) then we obtain a parameterized problem lying in FPT (see Corollary 1.20 in [28]):

p-deg-Dominating Set:

Instance:	A graph G and $k \in \mathbb{N}$.
Parameter:	$k + \deg(G)$.
Problem:	Decide whether G has a dominating set of cardinality k.

We end this brief introduction to fixed-parameter tractability with a standard technique to prove that a parameterized problem is fixed-parameter tractable. We use this technique to prove our FPT results.

Definition 2.2.5 ([28], Definition 1.38)**.** *Let (Q, κ) be a parameterized problem over Σ.*

*A polynomial time computable function $K\colon \Sigma^* \to \Sigma^*$ is a **kernelization** of (Q, κ) if there is a computable function $h\colon \mathbb{N} \to \mathbb{N}$ such that for all $x \in \Sigma^*$ we have*

$$(x \in Q \iff K(x) \in Q) \quad \text{and} \quad |K(x)| \leq h(\kappa(x)) \,.$$

*If K is a kernelization of (Q, κ), then for every instance x of Q the image $K(x)$ is called the **kernel** of x (under K).*

This means that a kernel is an equivalent instance of a parameterized problem the size of which is bounded by a function that only depends on the parameter. One can weaken Definition 2.2.5 by not requiring to obtain an instance of the same problem, but a, loosely speaking, related problem.

Definition 2.2.6 ([6], Definition 2)**.** *A **generalized kernelization algorithm** from a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ to another parameterized problem $L' \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that given $(x, k) \in \Sigma^* \times \mathbb{N}$, outputs in $p(|x| + k)$ time a pair $(x', k') \in \Sigma^* \times \mathbb{N}$ such that*

(i) $(x, k) \in L \iff (x', k') \in L'$,

(ii) $|x'|, k' \leq f(k)$,

where f is an arbitrary computable function, and p a polynomial.

Analogously to Definition 2.2.5, we call the image of a generalized kernelization algorithm a **generalized kernel**. The concept of generalized kernelization is also known as bikernelization [2].

If we apply a brute force algorithm to a (generalized) kernel, then its running time depends only on the parameter. This fact is the idea to prove the sufficiency-part of the following theorem. Furthermore, this is also the reason why it is demanded that the construction of the kernel must be computable in polynomial time since this part is usually depending on the original instance size.

Theorem 2.2.7 ([28], Theorem 1.39)**.** *For every parameterized problem (Q, κ), the following are equivalent:*

(i) $(Q, \kappa) \in FPT$.

(ii) Q is decidable, and (Q, κ) has a kernelization.

Theorem 2.2.7 also holds for generalized kernelization, which can be proved in the same way. We apply this result in Chapter 5. First, we reduce the size of a given graph in two ways. Then, we transform the relevant properties to another mathematical object the size of which is – after another reduction – depending only on the considered parameters. The exact procedure is given by so-called **reduction rules**.

For an extensive study of parameterized complexity we recommend the books of Downey and Fellows [25] and Flum and Grohe [28].

Chapter 3

Equidomination

In Section 3.1, we introduce the topic of equidomination. We obtain several general results regarding standard graph classes and operations in Section 3.2. In Section 3.3, we characterize the class of hereditarily equidominating graphs and give a recognition algorithm that uses a decomposition result.

3.1 An Introduction

The concept of equidomination was introduced by Payan in 1980 [56]. Three years before, Benzaken and Hammer [3] defined a graph to be **domishold** if there are positive weights associated to the vertices of the graph such that a subset D of vertices is dominating if and only if the sum of the weights of the vertices of D exceeds a certain threshold. In other words, the characteristic vectors of dominating sets are exactly the zero-one solutions of a linear inequality, where the coefficients of the inequality correspond to the weights of the vertices and the right-hand side to the threshold. The definition of domishold graphs led Payan to pose the following questions [56]:

> "Which are among domishold graphs those for which *minimal* dominating sets are characterized by the 0,1-solutions of a linear *equation*."

This question motivated Payan to define equidominating graphs:

Definition 3.1.1. *A graph $G = (V, E)$ is called **equidominating** if there exists $t \in \mathbb{N}$ and a weight function $\omega\colon V \to \mathbb{N}$ such that for all $D \subseteq V$ the following equivalence holds:*

$$D \text{ is a minimal dominating set} \iff \omega(D) = t\,.$$

*Further, we call the pair (ω, t) an **equidominating structure**, ω an **equidominating function** and t a **target value**.*

We point out again that a minimal dominating set is meant to be inclusion-wise minimal dominating. That is the removal of any vertex of a minimal dominating set results in a non-dominating set. Equivalently, every vertex of a minimal dominating set has at least one private neighbor.

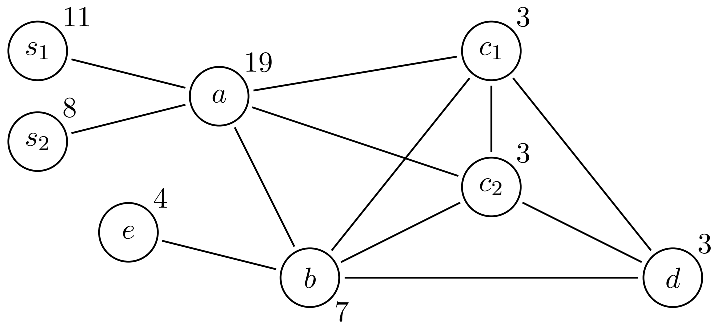


Figure 3.1: An equidominating graph on 8 vertices; the weights are drawn next to the vertices and the target value is $t = 26$.

Even though the term equidominatable might be more suitable, we will use equidominating for historical reasons. Note that $0 \notin \mathbb{N}$. Thus, every vertex has weight at least one. As we will see, the sufficiency-part in the definition of equidominating – ensuring that no other subset except minimal dominating sets has a total weight t – is what often makes things complicated.

Equivalently, we can define a graph to be equidominating if there is a weight function with real-valued weights $\omega\colon V \to \mathbb{R}$ such that $D \subseteq V$ is a minimal dominating set if and only if $\omega(D) = 1$. However, for convenience we will work with integer weights. Figure 3.1 shows an equidominating graph. Every minimal dominating set has total weight 26 and further, every subset of total weight 26 is a minimal dominating set: take for example a look at the minimal dominating sets $\{a, d, e\}$ and $\{b, s_1, s_2\}$, the non-minimal dominating set $\{a, b, d\}$ or the non-dominating set $\{c_1, e\}$.

We can also interpret the concepts geometrically. As initially motivated by Payan, the weights of an equidominating function can be understoodd as coefficients of a linear equality with the target value as the right-hand side. This means that the characteristic vectors of minimal dominating sets of an equidominating graph lie in one hyperplane. Furthermore, for domishold graphs there is a hyperplane that separates the characteristic vectors of dominating sets from those of non-dominating sets.

In his paper, Payan could answer the above stated question by the following characterization.

Theorem 3.1.2 ([56], Theorem 2)**.** *The following statements are equivalent:*

(i) G is a domishold and an equidominating graph.

(ii) G is a domishold and an equidominating graph and moreover the same threshold t_G and the same mapping ω_G can be used to characterize dominating sets by inequality and minimal dominating sets by equality.

(iii) G can be built from the graph with one vertex or from the complement of a matching, by repeated addition of isolated and dominating vertices.

(iv) G has no induced subgraph isomorphic to $2K_2$, P_3, $K_{3,2}$ or $\overline{P_2 \cup P_3}$.

We will see that the complement of a (perfect) matching, as mentioned by Payan in (iii), appears several times in the course of this thesis. For convenience, we will refer to it as $T(2n, n)$ and to a perfect matching on $2n$ vertices as $\overline{T(2n, n)}$ ($n \in \mathbb{N}$). As an induced subgraph, $T(2n, n)$ will play a crucial role in Chapter 4 and Chapter 5.

The analogue of domishold graphs for stable sets are called **threshold** graphs: a subset of vertices is a stable set if and only if the total weight is less than or equal to a threshold (see for example [44]). As threshold graphs satisfy (iii) and (iv) of Therorem 3.1.2 (see [17]), Payan concluded that threshold graphs are equidominating. However, besides Theorem 3.1.2 there exist no other results regarding characterizations of the class of equidominating graphs. In particular, there is no direct relation between domishold and equidominating graphs. There are equidominating graphs that are not domishold (for example $2K_2$) and vice versa (for example $K_{2,3}$, [44]). The next theorem shows that there is no characterization of equidominating graphs in terms of forbidden induced subgraphs.

Theorem 3.1.3. *Every graph can appear as an induced subgraph of an equidominating graph.*

Proof. Let $G = (V, E)$ be an arbitrary graph with vertex set $V = \{v_1, \dots, v_n\}$. We create a graph G' by adding an pendant vertex to every vertex of G, that is

$$G' = \big(V', E'\big) := \big(V \cup \{v'_1, \dots, v'_n\}, E \cup \{(v_i, v'_i) \mid i = 1, \dots, n\}\big) \ .$$

The graph G' is the so-called corona of G and K_1. Since we did not add edges between the vertices $\{v_1, \dots, v_n\}$, G clearly is an induced subgraph of G'.

To see that G' is equidominating we define the function $\omega\colon V' \to \mathbb{N}$ by $\omega(v_i) = \omega(v'_i) = 2^{i-1}$ for $i = 1, \dots, n$. Now, since every minimal dominating set contains, for each $i \in [n]$, either v_i or v'_i, every minimal dominating set has a total weight $2^n - 1$. Moreover, by considering the binary numeral system one can see that the weights of a subset X only sum up to $2^n - 1$ if $|X \cap \{v_i, v'_i\}| = 1$ for all $i \in [n]$. Thus, G' is equidominating with the equidominating structure $(\omega, 2^n - 1)$. □

On the one hand, this means that the class of equidominating graphs is not hereditary (compare Section 3.3). On the other hand, the question of determining the minimal hereditary class containing the class of equidominating graphs is void since this is the class of all graphs.

3.2 General Results

The following observation follows immediately from the definition of equidominating graphs.

Observation 3.2.1. *Let G be an equidominating graph with equidominating structure (ω, t). Then $(c\omega, ct)$ is an equidominating structure of G for all $c \in \mathbb{N}$.*

Of course, we can also use a factor $c \in \mathbb{Q}$ if we can ensure that we only obtain positive, integer values.

The next theorem deals with some basic graph classes and the question whether they are equidominating or not. Even though some of the statements might appear trivial, they are listed here for the sake of completeness. Moreover, they help to get a feeling for equidomination.

Theorem 3.2.2. *Let m, $n \in \mathbb{N}$.*

(i) The graphs K_n, $\overline{K_n}$, $K_{1,n}$, $T(2n, n)$ and $\overline{T(2n, n)}$ are equidominating.

(ii) The graph $K_{m,n}$ is equidominating if and only if $m = 1$ or $n = 1$ or $m = n = 2$.

(iii) The path P_n is equidominating if and only if $n \le 4$.

(iv) The cycle C_n is equidominating if and only if $n \le 4$.

Proof. Let m, $n \in \mathbb{N}$.

(i) Let ω^* be the weight function that is constant equal to 1. Then $(\omega^*, 1)$, (ω^*, n) and $(\omega^*, 2)$ are equidominating structures of K_n, $\overline{K_n}$ and $T(2n, n)$, respectively. For $v \in K_{1,n}$ we define $\omega(v) \coloneqq \deg(v)$, then (ω, n) is an equidominating structure of $K_{1,n}$. The graph $\overline{T(2n, n)}$ is equidominating since it is the corona of $\overline{K_n}$ and K_1 (compare the proof of Theorem 3.1.3).

(ii) Following (i) and due to symmetry it remains to show that $K_{m,n}$ is not equidominating if $m \ge 2$ and $n \ge 3$ (note that $K_{2,2} = T(4, 2)$). Let V and W be the color classes of size m and n, respectively. Since for all $v \in V$ and $w \in W$ the set $\{v, w\}$ is a minimal dominating set, every equidominating function must be constant on V and on W. Further, V and W both are minimal dominating sets. Taken together, for any target value $t \in \mathbb{N}$ every vertex of V and W must have weight t/m and t/n, respectively. But now we have $\omega(\{v, w\}) = \frac{t}{m} + \frac{t}{n} < t$, a contradiction.

(iii) It follows from (i) and Theorem 3.1.3 that P_n is equidominating for $n = 1, \ldots, 4$ (note that P_4 is K_2 with a pendant vertex added to each vertex). So let $n \ge 5$, $V(P_n) = \{v_1, \ldots, v_n\}$, $E(P_n) = \{(v_i, v_{i+1}) \mid i = 1, \ldots, n-1\}$ and suppose that $\{\omega, t\}$ is an equidominating structure. For $k = 1, 2$ the sets $\{v_k\} \cup \{v_{2i} \mid 2 \le i \le \lfloor \frac{n}{2} \rfloor\}$ both are minimal dominating sets, so we get $\omega(v_1) = \omega(v_2)$. Moreover, $\{v_2\} \cup \{v_{1+2i} \mid 2 \le i \le \lfloor \frac{n}{2} \rfloor\}$ is a minimal dominating set and thus has a total weight t. Now, $\{v_1\} \cup \{v_{1+2i} \mid 2 \leq i \leq \lfloor \frac{n}{2} \rfloor\}$ also has a total weight t, but v_3 is not dominated, a contradiction.

(iv) By (i), C_n is equidominating for $n = 3, 4$ (note that $C_4 = T(4,2)$). So, let $n \geq 5$ and suppose that C_n is equidominating with equidominating structure $\{\omega, t\}$. Analogously to the proof of (iii), we can show that two vertices have the same weight and hence, due to the symmetry of C_n, all vertices. So, ω is constant on C_n. Since every minimal dominating set of C_n has $\lfloor \frac{n}{2} \rfloor$ vertices, $\omega \equiv \frac{t}{\lfloor \frac{n}{2} \rfloor}$. But if we take $\lfloor \frac{n}{2} \rfloor$ consecutive vertices of C_n, we obtain a subset of total weight t, which is clearly not a minimal dominating set, a contradiction.

□

We will now take a look at some graph operations and their relation to equidomination.

Theorem 3.2.3. *Let G be a graph and G' the graph obtained from G by adding a universal vertex. Then G is equidominating if and only if G' is equidominating.*

Proof. Let the graphs G and G' be as described above and let $x \in V(G')$ be the added, universal vertex. Let $\mathcal{D}$ and $\mathcal{D}'$ be the set of minimal dominating sets of G and G', respectively. Clearly, $\mathcal{D}' = \mathcal{D} \cup \{x\}$. Assume that G is equidominating and let (ω, t) be an equidominating structure of G. We define $\omega'\big|_{V(G)} :\equiv \omega$ and $\omega'(x) := t$. Then (ω', t) is an equidominating structure of G'.

Now assume that G' is equidominating and let (ω', t') be an equidominating structure of G. Let $\omega :\equiv \omega'\big|_{V(G)}$. Then (ω, t') is an equidominating structure of G. □

The same holds for an isolated vertex, which can be proved similarly.

Theorem 3.2.4. *Let G be a graph and G' the graph obtained from G by adding an isolated vertex. Then G is equidominating if and only if G' is equidominating.*

Proof. Let the graphs G and G' be as described above and let $x \in V(G')$ be the added, isolated vertex. Let $\mathcal{D}$ and $\mathcal{D}'$ be the set of minimal dominating sets of G and G', respectively. The vertex x is contained in every minimal dominating set of G' and thus $\mathcal{D}' = \{D \cup \{x\} \mid D \in \mathcal{D}\}$.

Let (ω, t) an equidominating structure of G. We define a weight function ω' on $V(G')$ by $\omega'\big|_{V(G)} :\equiv \omega$ and $\omega'(x) := \omega(V(G))$. Further, let $t' := t + \omega(V(G))$. Then (ω', t') is an equidominating structure of G'.

Conversely, if (ω', t') is an equidominating structure of G', then (ω, t) defined by $\omega :\equiv \omega'\big|_{V(G)}$ and $t := t' - \omega'(x)$ is an equidominating structure of G. □

In the following remark, we summarize some operations that are not compatible with equidomination.

Remark 3.2.5. *(i) The complete union of two equidominating graphs is not equidominating in general. By 3.2.2(i), the graphs* $\overline{K_2}$ *and* $\overline{K_3}$ *are equidominating. However, the complete union of those graphs is* $K_{2,3}$ *which is not equidominating, by Theorem 3.2.2(ii).*

(ii) Two non-adjacent vertices that have the same open neighborhood are called false twins, while two vertices with identical closed neighborhood are called true twins (see Definition 4.3.1 on page 39 for a more detailed definition). Adding a false or a true twin does not preserve equidomination. If we add a false twin to the universal vertex of $K_{1,3}$*, we obtain the non-equidominating graph* $K_{2,3}$*. Further, adding a true twin to any vertex of* C_4 *yields the non-equidominating graph house (compare Lemma 3.3.3 below on page 26). It follows that substituting an equidominating graph for a vertex of another equidominating graph (see [16] for a definition) also does not preserve equidomination.*

(iii) The following graph products are standard graph-theoretic notation, for a definition see for example [40]. The Tensor product of two equidominating graphs is not necessarily equidominating. For example, C_6*, which is not equidominating by Theorem 3.2.2(iv), is the tensor product of the equidominating graphs* K_2 *and* C_3*. Furthermore, the Cartesian and the Strong product do not preserve equidomination, since both* $P_2 \,\square\, P_3$ *and* $P_2 \boxtimes P_4$ *are not equidominating, which can be shown similarly to the proof of Theorem 3.2.2(iii).*

(iv) Adding a pendant vertex to an equidominating graph does not preserve equidominating. This follows from Theorem 3.2.2(iii) since P_5 *can be obtained by adding a pendant vertex to either of the two vertices of degree 1 of the equidominating graph* P_4*.*

The next theorem seems to deal with an unusual operation. However, this operation is a generalization of the disjoint union and the chain-join of two graphs. Therefore, Theorem 3.2.6 can be used to deduce Corollary 3.2.7 and Corollary 3.2.8.

Theorem 3.2.6. *For* $i \in \{1,2\}$*, let* $G_i = (V_i, E_i)$ *be two disjoint, equidominating graphs and let* $U_i \subseteq V_i$ *be the (possibly empty) set of universal vertices of* G_i*. Further, let* $V_i \setminus U_i \neq \emptyset$*, for* $i \in \{1,2\}$ *(this means that* G_i *is not a complete graph).*

Let G *be the graph obtained by connecting the vertices of* U_1 *with the vertices of* U_2 *in an arbitrary way, that is*

$$G = \left(V_1 \cup V_2, (E_1 \cup E_2) \cup E'\right),$$

with $E' \subseteq U_1 \times U_2$*. Then* G *is equidominating.*

Proof. Let (ω_i, t_i) be an equidominating structure of G_i, for $i \in \{1,2\}$, and let $k = 1 + \sum_{v \in V_1} \omega_1(v)$. We define a weight function ω on G by $\omega\big|_{V_1} :\equiv \omega_1$ and $\omega\big|_{V_2} :\equiv k \cdot \omega_2$ and $t \coloneqq t_1 + kt_2$. We claim that (ω, t) is an equidominating structure of G.

To see this, pick any set $X \subseteq V(G)$ with $\omega(X) = t$. By the definition of ω and t, we get $\omega_1(X \cap V_1) = t_1$ and $\omega_2(X \cap V_2) = t_2$. Hence, $X_i \coloneqq X \cap V_i$ is a minimal dominating set of G_i, for $i \in \{1,2\}$. It follows that X is a dominating set of G. We have to show that X is minimal.

Suppose that X is not minimal, so there is some $x \in X$ with $N[X \setminus \{x\}] = V(G)$. Without loss of generality let $x \in X_1$. As X_1 is a minimal dominating set of G_1, there is some $y \in V_1$ not dominated by $X_1 \setminus \{x\}$. It follows that $y \in N_G[X_2]$ and hence $y \in U_1$. Since y is universal in G_1 and not dominated by $X_1 \setminus \{x\}$, we get $X_1 = \{x\}$. Consequently, $N_G[X_2] = V(G)$, a contradiction to $V_1 \setminus U_1 \neq \emptyset$. This shows that X is a minimal dominating set of G.

Conversely, let D be a minimal dominating set of G. We have to prove that $\omega(D) = t$. It suffices to show that $D_i \coloneqq D \cap V_i$ is a minimal dominating set of G_i, for $i \in \{1,2\}$. Then, $\omega_i(D_i) = t_i$, for $i \in \{1,2\}$, and so $\omega(D) = t_1 + kt_2 = t$.

By symmetry, it suffices to show that D_1 is a minimal dominating set of G_1. Since $V_1 \setminus U_1 \neq \emptyset$, we get $V_1 \not\subseteq N_G[V_2]$ and hence $D_1 \neq \emptyset$. If $U_1 \cap D_1 \neq \emptyset$, together with $V_2 \setminus U_2 \neq \emptyset$ it follows that $|D_1| = 1$. Thus, D_1 is a minimal dominating set of G_1. So we may assume that $U_1 \cap D_1 = \emptyset$.

Since $N_G[V_2] \cap (V_1 \setminus U_1) = \emptyset$, D_1 is a dominating set of $G_1 - U_1$ and thus of G_1. Suppose that D_1 is not minimal. Then there is some $x \in D_1$ such that $D_1' \coloneqq D_1 \setminus \{x\}$ is still a dominating set of G_1. As $U_1 \cap D_1 = \emptyset$, we have $N_G[D_1] \cap V_2 = \emptyset$. Hence, D_2 is a dominating set of G_2, and so $D_1' \cup D_2$ is a dominating set of G. However, this contradicts the minimality of D. This completes the proof. □

Indeed, in the preceding proof the factor $k = \omega(V_1)$ would be sufficient for most graphs G_1. However, if G_1 is an edgeless graph and thus $t_1 = \omega_1(V_1)$, and if there is a vertex $v \in G_2$ with $\omega_2(v) = 1$, then the argumentation fails. Therefore, we add 1 in the definition of k.

Furthermore, to prove Theorem 3.2.6 we need that every minimal dominating set of G is a disjoint union of two minimal dominating set of the graphs G_1 and G_2. Since this is not necessarily the case if one of the graphs only has universal vertices, we have to except complete graphs.

As we do not require to add any edge in Theorem 3.2.6, we obtain the following corollary.

Corollary 3.2.7. *The disjoint union of two equidominating graphs is equidominating.*

Proof. The only missing case is one of the graphs being a complete graph. However, this can be shown straightforwardly using the weight function ω and target value t defined in the proof of Theorem 3.2.6. □

Recall that a chain-join of two disjoint graphs connects the universal vertices of both graphs in such a manner, that afterward there is an inclusion chain between the neighborhoods of the (former) universal vertices of each graph. This operation is essential in the context of hereditarily equidominating graphs which we deal with in Section 3.3.

Corollary 3.2.8. *The chain-join of two equidominating graphs is equidominating.*

Proof. Again, it remains to show that the chain-join G of two equidominating graphs G_1 and G_2 is equidominating in the case that one of the graphs G_1 and G_2 is a complete graph. By Theorem 3.2.3, we may assume that G has no universal vertex. Then, by the definition of a chain graph, it follows that there are vertices $v_1 \in V(G_1)$ with $v_1 \notin N_G(V(G_2))$ and $v_2 \in V(G_2)$ with $v_2 \notin N_G(V(G_1))$. With this fact, the rest can be done analogously to the proof of Theorem 3.2.6. □

The next theorem is related to previous results. It is the generalization of Theorem 3.2.6 for several equidominating graphs.

Theorem 3.2.9. *The graph obtained by connecting the universal vertices of $n \in \mathbb{N}$ disjoint, non-complete, equidominating graphs in an arbitrary manner is equidominating.*

Proof. Let $n \in \mathbb{N}$. For $i \in [n]$, let G_i be an equidominating graph with equidominating structure (ω_i, t_i) and $U_i \subseteq V(G_i)$ the set of universal vertices of G_i. Further, let G be the graph obtained by connecting the vertices of the sets U_i in an arbitrary way.

If we define a weight function ω on $V(G)$ by $\omega\big|_{V(G_i)} :\equiv \omega_i$, then every minimal dominating set of G has a total weight $\sum_{i=1}^{n} t_i$. However, we also must ensure that no other subset of vertices has this total weight. The main idea to do this is to multiply the given equidominating structures by relatively large values such that subsets of different graphs cannot have the same total weight (compare the proofs of Theorem 3.1.3 and Theorem 3.2.6). This yields further equidominating structures of the graphs $G_1, \ldots, G_n$, by Observation 3.2.1.

To obtain the needed multipliers we define $k_1 := 1$ and $k_i := 1+\sum_{j=1}^{i} k_j \Big(\omega_j\big(V(G_j)\big)\Big)$ for $i = 2, \ldots, n$. Now, let $\omega\colon V(G) \to \mathbb{N}$ be the weight function defined by $\omega\big|_{V(G_i)} :\equiv k_i\omega_i$ $(i \in [n])$ and $t := \sum_{i=1}^{n} k_i t_i$. Following the discussion above, (ω, t) is an equidominating structure of G. □

Roughly speaking, the following theorem is about attaching an equidominating graph to every vertex of a graph.

Theorem 3.2.10. *For $n \in \mathbb{N}$, let G be a graph on the vertices $v_1, \ldots, v_n$ and $G_1, \ldots, G_n$ equidominating graphs. Further, let H be the graph obtained by connecting v_i to all vertices of G_i, for all $i = 1, \ldots n$, that is*

$$V(H) = V(G) \cup V(G_1) \cup \ldots \cup V(G_n)$$
$$E(H) = E(G) \cup E(G_1) \cup \ldots \cup E(G_n) \cup \{v_i w \mid w \in V(G_i),\ i = 1, \ldots n\}$$

Then H is equidominating.

Proof. We want to apply Theorem 3.2.9. For that we add a universal vertex u_i to every graph G_i, and then connect two vertices u_i and u_j if and only if $v_i v_j \in E(G)$ ($i, j \in [n]$). By this, we obtain a graph which is isomorphic to H and which is equidominating by Theorem 3.2.3 and Theorem 3.2.9. □

As in Theorem 3.1.3, it is important for Theorem 3.2.10 that we attach an equidominating graph to every vertex of G. Only then every minimal dominating set of H contains either v_i or a minimal dominating set of G_i, for every $i \in [n]$.

With the results of this section we get several operations that preserve equidomination:

- adding/deleting a universal vertex (Theorem 3.2.3),
- adding/deleting an isolated vertex (Theorem 3.2.4),
- connecting the universal vertices of non-complete graphs (Theorem 3.2.9),
- disjoint union (Corollary 3.2.7),
- chain-join (Corollary 3.2.8),
- attaching equidominating graphs (Theorem 3.2.10).

With these operations, we can construct more complicated equidominating graphs than those mentioned in Theorem 3.2.2. The graph of Figure 3.1 can also be obtained with these operations. Further research could be concerned with finding more operations preserving equidominating. An interesting question in this direction is if we can find an alternative definition of equidominating graphs as the graphs that can be constructed using such operations. Such definitions exist for other graph classes (for example threshold graphs, series-parallel graphs or cographs, see [10]). We will give such a characterization for so-called hereditarily equidominating graphs in Subsection 3.3.1 (see Theorem 3.3.2(c)).

3.3 Hereditarily Equidominating Graphs

This section deals with the class of hereditarily equidominating graphs. A graph G is called **hereditarily equidominating** if every induced subgraph of G is equidominating, that is if $G[X]$ is equidominating for all $X \subseteq V$. In Subsection 3.3.1 we give a characterization of the class of hereditarily equidominating graphs in terms of the list of forbidden induced subgraphs and a structural decomposition. This decomposition yields an $O(n(n+m))$ time recognition algorithm which we present in Subsection 3.3.2.

3.3.1 Forbidden Induced Subgraph Characterization and Structural Decomposition

Regarding our decomposition theorem below, we define the class of **basic graphs** as $\{K_1\} \cup \{T(2n,n) \mid n \in \mathbb{N}\}$. Remember that the Turán graph $T(2n,n)$ is the complement of a perfect matching on $2n$ vertices. By Theorem 3.2.2(i), we already know that every basic graph is equidominating. By the following lemma, we see that basic graphs are in fact hereditarily equidominating.

Lemma 3.3.1. *Every basic graph is hereditarily equidominating.*

Proof. For K_1 there is nothing to do. Let $n \geq 2$, then $T(2n,n)$ is equidominating, by Theorem 3.2.2(i). Further, every induced subgraph of $T(2n,n)$ is either $T(2n',n')$ or $T(2n',n')$ with a additional universal vertices ($n' \leq n$). Following Theorem 3.2.2(i) and 3.2.3, the induced subgraph is equidominating in both cases. □

Let H be a graph. A graph G is called ***H*-free** if no induced subgraph of G is isomorphic to H. For a set of graphs $\mathcal{H}$ we say that a graph G is **$\mathcal{H}$-free** if G is H-free for all $H \in \mathcal{H}$.

Let $\mathcal{C}$ be a graph class and G a graph. If there exists a set of graphs $\mathcal{H}$ such that $G \in \mathcal{C}$ if and only if G is $\mathcal{H}$-free, then $\mathcal{H}$ is said to be the **set of forbidden induced subgraphs** of $\mathcal{C}$.

We define $\mathcal{F} := \{P_5, C_5, \textit{bull}, \textit{banner}, \textit{house}, K_{2,3}, \overline{P_2 \cup P_3}\}$ (see Figure 3.2 for an illustration). As the next theorem shows, the set $\mathcal{F}$ is exactly the set of forbidden induced subgraphs of the class of hereditarily equidominating graphs.

Theorem 3.3.2. *For any graph G, the following assertions are equivalent.*

(a) G is hereditarily equidominating.

(b) G is $\mathcal{F}$-free.

(c) One of the following assertions holds.

(i) G is a basic graph.

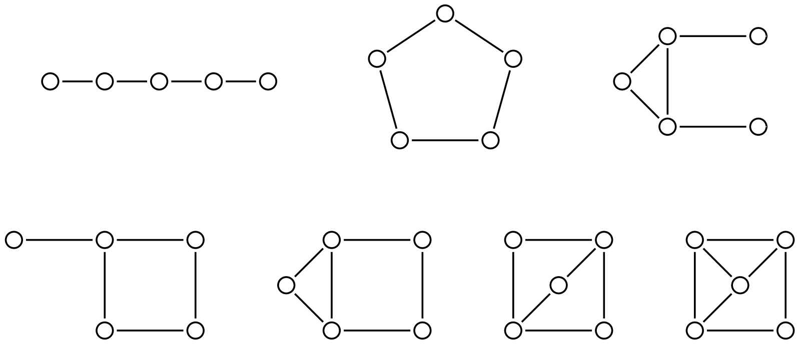


Figure 3.2: The set $\mathcal{F}$ of forbidden induced subgraphs of hereditarily equidominating graphs; top row: P_5, C_5, *bull*; bottom row: *banner*, *house*, $K_{2,3}$, $\overline{P_2 \cup P_3}$.

(ii) G is obtained from a hereditarily equidominating graph by adding a universal vertex.

(iii) G is the chain-join of two hereditarily equidominating graphs.

The proof of the above theorem builds upon the following five lemmas.

Lemma 3.3.3. *No graph in $\mathcal{F}$ is equidominating.*

Proof. By Theorem 3.2.2, we already know that P_5, C_5 and $K_{2,3}$ are not equidominating.

With an appropriate labeling of the vertices, we can prove that the remaining graphs are not equidominating with an identical argumentation.

For that let us denote the five vertices of each graph by a, b, c, d and e from left to right and top to bottom. For example, in the *bull* graph the vertex of degree 2 is denoted by a, the upper (lower) vertex of degree 3 by b (by c) and the upper (lower) vertex of degree 2 by d (by e).

Now, let us suppose any graph of *bull*, *banner*, *house* or $\overline{P_2 \cup P_3}$ is equidominating. Since $\{a, e\}$ and $\{b, e\}$ both are minimal dominating sets, a and b have the same weight. Furthermore, $\{b, c\}$ is a minimal dominating set. But now, $\{a, c\}$ has the same total weight as $\{b, c\}$ without being a minimal dominating set, a contradiction. □

We need a variant of Theorem 3.2.3 for hereditarily equidominating graphs.

Lemma 3.3.4. *Let G be a graph and G' the graph obtained from G by adding a universal vertex. Then G is hereditarily equidominating if and only if G' is hereditarily equidominating.*

Proof. Sufficiency is trivial. So, let G be hereditarily equidominating. Every induced subgraph H' of G' is either an induced subgraph of G or an induced subgraph of G with an additional universal vertex. In both cases is H' equidominating. In the first case since G is hereditarily equidominating and in the latter case due to Theorem 3.2.3. □

The next lemma shows that the chain-join operation preserves not only equidominating but also hereditarily equidominating graphs.

Lemma 3.3.5. *Let G_1 and G_2 be two hereditarily equidominating graphs, and let G be a chain-join of G_1 and G_2. Then G is hereditarily equidominating.*

Proof. By Corollary 3.2.8, G is equidominating. Let H be any induced subgraph of G. Then H is a chain-join of the two graphs $H_1 := G_1[V(H) \cap V(G_1)]$ and $H_2 := G_2[V(H) \cap V(G_2)]$. This is for two reasons. First, the universal vertices of G_1 and G_2 are (if existent) also universal in H_1 and H_2, respectively. Secondly, the neighborhood relation of two universal vertices does not change in any induced subgraph.

As G_1 and G_2 are hereditarily equidominating, H_1 and H_2 are equidominating. By applying Corollary 3.2.8, we see that H is equidominating, too. As H is arbitrary, G is hereditarily equidominating. □

We now come to the decomposition of $\mathcal{F}$-free graphs. A **connected dominating set** is a dominating set that induces a connected subgraph. A **minimal connected dominating** set is a connected dominated set that does not properly contain a connected dominating set.

Lemma 3.3.6. *Let G be a connected $\mathcal{F}$-free graph without a universal vertex. Assume that G has a minimal connected dominating set D of size two, say $D = \{x, y\}$, such that there are private neighbors $x' \in pn(x, D)$ and $y' \in pn(y, D)$ with $x'y' \in E(G)$. Then $G \cong T(2n, n)$ for some $n \geq 2$.*

Proof. Since $G[\{x, y, x', y'\}] \cong C_4$ and G is $\mathcal{F}$-free, the set $D' := \{x, y, x', y'\}$ is a module (recall that this means every vertex of $V(G) \setminus D'$ is either adjacent to every vertex of D' or to none). However, $D = \{x, y\}$ is a dominating set, and thus D' is a dominating set, too. So, D' is a module and a dominating set. Hence, G is the complete union of $G[D']$ and $G[V(G) - D']$.

Let us pick any vertex $z \in V(G) - D'$. Since G does not have a universal vertex, there must be some non-neighbor z' of z. Suppose that there is a second non-neighbor of z, say z''. Consider the graph $G' := G[\{x, y', z, z', z''\}]$. If $z'z'' \in E(G)$, then $G' \cong \overline{P_2 \cup P_3}$, a contradiction. Hence, $z'z'' \notin E(G)$, and so $G' \cong K_{2,3}$, again a contradiction.

Thus, every vertex of G has exactly one non-neighbor in G. This means that $G \cong T(2n, n)$, for some $n \geq 2$. □

The next lemma is crucial for the decomposition theorem. It also serves as a basis for the recognition algorithm presented in Section 3.3.2.

Lemma 3.3.7. *Let G be a connected $\mathcal{F}$-free graph without universal vertex that has a connected dominating set of size two. Assume that for every minimal connected dominating set D of size two, say $D = \{x, y\}$, it holds that $x'y' \notin E(G)$ for any two private neighbors $x' \in pn(x, D)$ and $y' \in pn(y, D)$. Then G is a chain-join of two disjoint connected $\mathcal{F}$-free graphs G_1 and G_2.*

Proof. Let $\{x, y\}$ be a connected dominating set such that $|N(u)| + |N(v)| \leq |N(x)| + |N(y)|$, for all connected dominating sets $\{u, v\}$ of G of size two. Let $X := N[x] \setminus N[y]$, $Y := N[y] \setminus N[x]$ and $S := N[x] \cap N[y]$. It is clear that X, Y and S are pairwise disjoint and $X \cup Y \cup S = V(G)$. Note that, as $\{x, y\}$ is a minimal connected dominating set, $X \neq \emptyset$ and $Y \neq \emptyset$.

Let $z \in S \setminus \{x, y\}$ be arbitrary. Since G does not have a universal vertex, there is a non-neighbor of z. Suppose first that $X \cup Y \subseteq N(z)$. Then there is some $z' \in S$ with $zz' \notin E(G)$. Since G is *bull*-free, z' has some neighbor in $X \cup Y$, say $x' \in X \cap N(z)$. Choose any $y' \in Y$. If $y'z' \in E(G)$, then $G[\{x', y, y', z, z'\}] \cong \overline{P_2 \cup P_3}$, and if otherwise $y'z' \notin E(G)$, then $G[\{x', y, y', z, z'\}] \cong$ *house*. Since both are contradictory, we know that z has some non-neighbor among $X \cup Y$, say in Y. As G is *bull*-free, z is adjacent to all elements of X. In particular, $\{z, y\}$ is a connected dominating set.

Suppose that there is some $y' \in N(z) \cap Y$. We claim that $|N(z)| > |N(x)|$, in contradiction to our assumption that, among all dominating sets of G of size two, $|N(x)| + |N(y)|$ is maximum. To prove this claim, suppose $|N(z)| \leq |N(x)|$. Since $X \cup \{y'\} \subseteq N(z)$ and $y' \notin N(x)$, there must be some $z' \in S \setminus N(z)$. Pick any $x' \in X$. Since G is *bull*-free, $x'z' \in E(G)$ or $y'z' \in E(G)$. Without loss of generality let $x'z' \in E(G)$. Like above, this leads to $G[\{x', y, y', z, z'\}] \cong \overline{P_2 \cup P_3}$ or $G[\{x', y, y', z, z'\}] \cong$ *house*, depending on whether $y'z' \in E(G)$ or not. Since both is contradictory, we obtain $N(z) \cap Y = \emptyset$.

Summing up, $X \subseteq N(z)$ and $N(z) \cap Y = \emptyset$. As z is arbitrary, for every vertex $z' \in S$ it either holds that $X \subseteq N(z')$ and $N(z') \cap Y = \emptyset$, or $Y \subseteq N(z')$ and $N(z') \cap X = \emptyset$. This partitions the set S into two disjoint sets X' and Y' where every element of X' is adjacent to X and every element of Y' is adjacent to Y. Consider the bipartite graph B with color classes X' and Y' induced by the edge set $E_G(X', Y') := \{xy \in E(G) \mid x \in X', y \in Y'\}$. Let G_1 and G_2 be the connected components of $G - E_G(X', Y')$ containing X and Y, respectively. Note that both G_1 and G_2 are induced subgraphs of G and hence, also $\mathcal{F}$-free. Since the elements of X' are universal vertices of G_1 and the elements of Y' are universal vertices of G_2, it remains to prove that B is a chain graph. Then G is a chain-join of G_1 and G_2 and we are done.

Suppose the opposite holds. That means there are vertices x', $x'' \in X' \cup \{x\}$ and y', $y'' \in Y' \cup \{y\}$ such that $x'y'$, $x''y'' \in E(B)$ and $x'y''$, $x''y' \notin E(B)$. Let $x''' \in X$. Then $G[\{x', x'', x''', y', y''\}] \cong$ *house*, a contradiction. This completes the proof. □

Now, everything is prepared to prove the decomposition theorem. We prove that (a) follows from (c) secondly, as we use it to show that (c) follows from (b).

Proof of Theorem 3.3.2. (a)$\Longrightarrow$(b): Follows from Lemma 3.3.3.

(c)$\Longrightarrow$(a): Now, let G be such that one of the following conditions holds.

(i) G is a basic graph.

(ii) G is obtained from a hereditarily equidominating graph by adding a universal vertex.

(iii) G is the chain-join of two hereditarily equidominating graphs.

If G is basic, it is hereditarily equidominating by Theorem 3.2.2. By Lemma 3.3.4 and Lemma 3.3.5, it follows that any graph obtained from a hereditarily equidominating graph by attaching a universal vertex or from two hereditarily equidominating graphs by a chain-join is hereditarily equidominating, too.

(b)$\Longrightarrow$(c): Finally, let G be an $\mathcal{F}$-free graph. We may assume that G is not basic.

Suppose that G has a universal vertex x and let $G' := G - x$. Clearly, G' is $\mathcal{F}$-free, too. If we can show that (b) implies (c) for G', then – since we already know that (c) implies (a) – we get that G' is hereditarily equidominating and hence, that G satisfies (c). Moreover, since the disjoint union is a particular chain join, the same argumentation holds if G' is a connected component of G. This means that we have to prove that (b) implies (c) for every connected component of G without any universal vertex (regarding the connected component).

So, let H be a connected component of G without a universal vertex. Let D be any minimal connected dominating set of H. Since H is $\{P_5, C_5\}$-free and connected, D is a clique [12]. Suppose that $|D| \geq 3$, and let x, y, $z \in D$ be distinct vertices. Let x' be a private neighbor of x, and y' be a private neighbor of y. If $x'y' \notin E(H)$, then $H[\{x, x', y, y', z\}] \cong$ *bull*, a contradiction. Thus $x'y' \in E(H)$. But then $H[\{x, x', y, y', z\}] \cong$ *house*, another contradiction. This shows that $|D| \leq 2$.

As we assumed that H does not have a universal vertex, we get $|D| = 2$, say $D = \{x, y\}$. If x and y have private neighbors, say $x' \in pn(x, D)$ and $y' \in pn(y, D)$, such that $x'y' \in E(H)$, then H is basic by Lemma 3.3.6.

So we may assume that for every minimal connected dominating set $D = \{x, y\}$ of H, every private neighbor of x is non-adjacent to every private neighbor of y. Hence, H is the chain-join of two $\mathcal{F}$-free graphs H_1 and H_2, by Lemma 3.3.7.

It remains to show that H_1 and H_2 are hereditarily equidominating. Though, for that we can use the same argumentation as above. This completes the proof. $\square$

The proof of Theorem 3.3.2 indicates our recognition algorithm and its recursive character. We consider connected components without any universal vertex. Either these are basic or a chain-join of two graphs. In the latter case, we continue with those two graphs.

3.3.2 Recognition of Hereditarily Equidominating Graphs

Given the fact that hereditarily equidominating graphs admit a finite forbidden induced subgraph characterization, it is clear that this class can be recognized in polynomial time by a simple brute force approach. However, a faster recognition is possible using the decomposition provided by Theorem 3.3.2.

At the end of this subsection, we will also prove the existence of a linear time recognition algorithm. However, due to the applied meta-theorem using a variation of so-called monadic second-order logic, we only prove the existence here, without stating it explicitly.

Our (explicit) algorithm mimics the decomposition of Theorem 3.3.2 (in particular see the part of the proof where we show that (b) implies (c)). In order to detect whether a considered graph is a chain join, the algorithm basically follows the proof of Lemma 3.3.7.

Recall that a co-chain graph is the complement of a chain graph. Equivalently, a co-chain graph is a graph obtained from some chain graph by turning the two color classes into cliques. These cliques are called the **co-classes** of the co-chain graph. Note that the partition of a co-chain graph into its two co-classes might not be unique.

Theorem 3.3.8. *Let G be a graph on n vertices and m edges. It can be decided in time $\mathcal{O}(n(n+m))$ whether G is a hereditarily equidominating graph.*

Proof. Let G be a graph on n vertices and m edges. To decide whether G is hereditarily equidominating, we apply the steps presented below to every connected component of G.

For a connected graph H we proceed as follows. Here, H is either a connected component of G or a graph that occurs during the decomposition and to that we reapply the algorithm (compare Step 11). The input graph G is hereditarily equidominating if and only if the algorithm terminates without returning at any point that G is not hereditarily equidominating. For the sake of clarity, we omit the recursive frame of the algorithm.

1. Compute the degrees of all vertices of H.
2. If H is basic, return that H is hereditarily equidominating.
3. If H has a universal vertex, say v, reapply Step 1–11 to each connected component of $H - v$.

4. Let x be a vertex of maximum degree, and let v be any non-neighbor of x.
5. Let y be the vertex of maximum degree among $N_H(v)$.
6. Compute the sets $X := N_H(x) \setminus N_H[y]$, $Y := N_H(y) \setminus N_H[x]$, and $S := N_H[x] \cap N_H[y]$.
7. Compute the sets $X' := \{x' \in S : N_H(x') \cap X \neq \emptyset\}$ and $Y' := S \setminus X'$.
8. Compute the graph $H' := H[S]$.
9. Check whether H' is a co-chain graph with co-classes X' and Y'. If not, return that G is not hereditarily equidominating.
10. Check whether the following conditions are satisfied:
    - for every $x' \in X'$ it holds that $X \subseteq N_H(x')$ and $Y \cap N_H(x') = \emptyset$;
    - for every $y' \in Y'$ it holds that $Y \subseteq N_H(y')$ and $X \cap N_H(y') = \emptyset$.

    If one of these conditions fails to hold, return that G is not hereditarily equidominating.
11. Reapply the algorithm to each connected component of $H - S$.

Let us first show how the algorithm can be implemented such that each iteration runs in $\mathcal{O}(|V(H)| + |E(H)|)$ time. Since at least one vertex is removed in each iteration, the overall running time of the algorithm is of order $\mathcal{O}(n(n+m))$.

Note that $H \cong T(2k, k)$, for some $k \geq 2$, if and only if $|V(H)| \geq 4$ and every vertex of H has degree $|V(H)| - 2$. Hence, it can be checked in linear time whether H is basic. Since all other steps are standard, it remains to discuss how to perform Step 9. In this particular step, we have to decide whether H' is a co-chain graph with the prescribed co-classes X' and Y'. As shown by Heggernes and Kratsch [34], co-chain graphs can be recognized in time $\mathcal{O}(|V(H)| + |E(H)|)$. It is straightforward that the algorithm of Heggernes and Kratsch can be modified such that it includes prescribed co-classes. Hence, Step 9 can be performed in time $\mathcal{O}(|V(H)| + |E(H)|)$.

We now come to the correctness of the algorithm. Our aim is to show that the algorithm performs a decomposition according to Theorem 3.3.2 (if possible). For this, it suffices to show that in each iteration, the algorithm performs a single step of such a decomposition, or correctly decides that a decomposition is no longer possible.

Let us first show that if H is a hereditarily equidominating graph, the algorithm correctly decomposes H. By Theorem 3.3.2, H is either basic, has a dominating vertex or is a chain-join.

In the case that H is basic, which is checked in Step 2, the algorithm can safely return that H is hereditarily equidominating. If H has a universal vertex, say v, this is detected in Step 3. By Lemma 3.3.4, H is hereditarily equidominating if and only if $H - v$ is hereditarily equidominating. Hence, the algorithm is correctly reapplied to the connected components of $H - v$.

So let us assume that H is neither basic nor has a universal vertex. Then, according to Theorem 3.3.2, H is the chain-join of two graphs H_1 and H_2. Let U_i be the set of universal vertices of H_i, for $i \in \{1, 2\}$. Let x be a vertex of maximum degree in H. As H is a chain-join of H_1 and H_2, it holds that $x \in U_1 \cup U_2$, say $x \in U_1$. Moreover, x has the maximal closed neighborhood in H among all vertices of U_1.

Let $v \in V(H) \setminus N_H[x]$, and let y be the vertex of maximum degree among $N_H(v)$. Then $v \in V(H_2) \setminus N_H(x) = V(H_2) \setminus N_H(V(H_1))$. Hence, $y \in U_2$ and y has the maximal closed neighborhood in H among all vertices of U_2.

Now we define $X := N_H(x) \setminus N_H[y]$, $Y := N_H(y) \setminus N_H[x]$, $S := N_H[x] \cap N_H[y]$, $X' := \{x' \in S : N_H(x') \cap X \neq \emptyset\}$, and $Y' := S \setminus X'$. Since $X' \subseteq U_1$, $Y' \subseteq U_2$, and H is a chain-join, it holds that $H[S]$ is a co-chain graph with co-classes X' and Y'. Thus, both X' and Y' satisfy the conditions checked in Step 10. Note that the graph $H - S$ equals the disjoint union of $H_1 - X'$ and $H_2 - Y'$, both being hereditarily equidominating by assumption. Hence, the algorithm is correctly reapplied to the connected components of $H - S$.

Let us now assume that the algorithm performs one iteration on H and does not return that G is not hereditarily equidominating. We have to show that H is hereditarily equidominating if and only if all graphs are hereditarily equidominating to which the algorithm is reapplied. Clearly, this holds if H has a universal vertex. So, we proceed to analyze the case that the reapplication is called in Step 11.

In this case the sets S, X', Y' and the graph H' are computed, and both Step 9 and Step 10 are performed successfully. In particular, H' is a co-chain graph with co-classes X' and Y'. This means that X' and Y' are cliques in H' and thus in H, and that the graph induced by the edge set $E_H(X', Y') := \{xy \in E(H) \mid x \in X', y \in Y'\}$ is a chain graph. After Step 10, we know that for every $x' \in X'$ it holds that $X \subseteq N_H(x')$ and $Y \cap N_H(x') = \emptyset$, and that for every $y' \in Y'$ it holds that $Y \subseteq N_H(y')$ and $X \cap N_H(y') = \emptyset$. Hence, H is a chain-join of the two connected components of the graph $H - E_H(X', Y')$. Let us denote these two components by H_1 and H_2. Moreover, let $H_1' := H_1 - X'$ and $H_2' := H_2 - Y'$. Note that $H - S = H_1' \cup H_2'$.

By Theorem 3.3.2, H is hereditarily equidominating if and only if both H_1 and H_2 are hereditarily equidominating. Since X' is a set of universal vertices of H_1, H_1 is hereditarily equidominating if and only if H_1' is hereditarily equidominating, again by Theorem 3.3.2. The analogous statement holds for H_2 and H_2'. Summing up, H is hereditarily equidominating if and only if $H - S = H_1' \cup H_2'$ is hereditarily equidominating. This completes the proof. $\square$

Thanks to an anonymous reviewer of the publication process of the article version we can prove the existence of a linear time recognition algorithm for hereditarily equidominating graphs. For that, we need some notations which we introduce only briefly here. We give further references for all involved topics.

The following argumentation intersects with the wide field of propositional logic, in particular with the so-called **Monadic Second-Order Logic** (**MSOL**). A detailed consideration of this topic goes beyond the scope of this argumentation. For an introduction as well as a deeper insight the interested reader is referred to the book [19].

To catch a glimpse of the different variations of MSOL in the context of graph theory we state the following quotation from Courcelle et al. from the introduction of [20].

> "Roughly speaking, $MSOL(\tau_1)$ is Monadic Second-Order Logic with quantification over subsets of vertices, but not of edges; $MSOL(\tau_{1,p})$ is the extension of $MSOL(\tau_1)$ by unary predicates representing labels attached to the vertices. $LinEMSOL(\tau_{1,p})$ is the extension of $MSOL(\tau_{1,p})$ which allows one to search for sets of vertices which are optimal with respect to some linear evaluation function."

By Theorem 3.3.2, we have a characterization of hereditarily equidominating graphs in terms of finitely many forbidden induced subgraphs. This property is expressible in $LinEMSOL(\tau_{1,p})$ [18]. Hence, to prove the existence of a linear time recognition algorithm, we can use the following result by Courcelle et al.

Theorem 3.3.9 ([20], Theorem 4)**.** *Let $\mathcal{C}$ be a class of graphs of clique-width at most k such there is a $\mathcal{O}\big(f(|E|,|V|)\big)$ algorithm, which, for each graph $G = (V, E)$ in $\mathcal{C}$, constructs a k-expression defining it. Then every $LinEMSOL(\tau_{1,p})$ problem on $\mathcal{C}$ can be solved in time $\mathcal{O}\big(f(|E|,|V|)\big)$.*

Before we can fully understand this theorem we need to define the terms clique-width and k-expression. The **clique-width $cwd(G)$** of a graph G is defined by the minimum labels needed to construct G with the following four operations (compare [9]):

(i) creation of a vertex labeled by integer i,

(ii) disjoint union of two labeled graphs,

(iii) join between all vertices with label i and all vertices with label j for $i \neq j$,

(iv) relabeling all vertices of label i be label j.

Furthermore, let G be a graph of clique-width k. A ***k*-expression** of G is an expression to construct G with the above mentioned operations using not more than k different labels. For complete and more precise definitions of clique-width and k-expression see for example [21].

We need the next lemma to show that we can obtain a k-expression in linear time for a given graph we want to check for being hereditarily equidominating. Then it follows – together with Theorem 3.3.2 and Theorem 3.3.9 – that hereditarily equidominating graphs are recognizable in linear time. Recall that a graph $G = (V, E)$ is prime if all its modules are trivial, that is if $|M| \in \{0, 1, |V|\}$ for all modules $M \subseteq V$ of G.

Lemma 3.3.10. *If G is a prime ($P_5, \overline{P_5}$, bull, banner)-free graph with $V(G) \geq 6$, then G is a co-chain graph.*

Proof. Theorem 1.2. of [29] implies that G or $\overline{G}$ is bipartite. Further, every prime P_5-free bipartite graph is a chain graph [53]. It follows – since every prime *banner*-free chain graph has at most four vertices – that G cannot be bipartite. Thus, $\overline{G}$ is bipartite. Since G is $\overline{P_5}$-free, $\overline{G}$ is P_5-free and hence a chain graph. This finishes the proof. □

Now, to decide if a given graph G is hereditarily equidominating we first apply modular decomposition to obtain all prime induced subgraphs. Then we check if every prime induced subgraph with more than five vertices is a co-chain graph. This can be done in linear time [34]. If we find a prime induced subgraph with more than five vertices that is not a co-chain graph, then it is not ($P_5, \overline{P_5}$, *bull*, *banner*)-free by Lemma 3.3.10. Hence, G is not ($P_5, \overline{P_5}$, *bull*, *banner*)-free and thus not hereditarily equidominating by Theorem 3.3.2.

If a prime induced subgraph of G is a chain-join, then it has clique-width four and a 4-expressions can be obtained in linear time (see Proposition 2 of [9]). If a prime induced subgraph of G has at most five vertices, then its clique-width is at most five and we can get a k-expression ($k \leq 5$) in constant time.

Finally, we can combine the k-expressions of all prime induced subgraphs of G to obtain a k-expression of G, again with the help of the modular decomposition [21]. Taken together we get the following theorem.

Theorem 3.3.11. *There exists a linear time algorithm that decides whether a given graph is hereditarily equidominating.*

Chapter 4

Complexity Issues and the Pseudo Class Partition

In this chapter, we will embed equidomination into the framework of complexity theory. First, we define and discuss the EQUIDOMINATION problem in Section 4.1. Then, we introduce two parameterized versions of the EQUIDOMINATION problem in Section 4.2. In Section 4.3, we elaborate a decomposition of a graph that we use to prove the fixed-parameter results in the next chapter. Using this decomposition, we state an algorithm to solve the parameterized versions of the EQUIDOMINATION problem in Section 4.4

4.1 The Equidomination Problem

We start with defining the following decision problem:

EQUIDOMINATION:

Instance: A graph G.
Problem: Decide whether G is equidominating.

The EQUIDOMINATION problem deals with the question whether a given graph is equidominating or not. Although the complexity of this problem is unknown, we firmly believe it is at least *NP*-hard. Apparently, it is not even known whether the EQUIDOMINATION problem is in *NP*. For a given function it is not easy to verify that it is an equidominating function. To do so, one has to calculate the total weight of every possible subset of vertices and check if exactly the minimal dominating sets obtain a specific total weight. Yet, we can give a complexity result regarding the problem defined by this question. We call this problem EQUIDOMINATING FUNCTION.

EQUIDOMINATING FUNCTION:

Instance: A graph $G = (V, E)$ and a function $\omega\colon V \to \mathbb{N}$.
Problem: Is ω an equidominating function?

Note that there is no target value predetermined. The next theorem and its proof are analogous to its equistable version discussed by Milanič et al. [49], where it is asked if a given function is an equistable function. Therefore, we will only sketch the proof here. To show their result, the authors work with a graph that is the disjoint union of several K_2. This graph is isomorphic to $\overline{T(2n,n)}$. For the graphs $\overline{T(2n,n)}$ ($n \in \mathbb{N}$), the family of minimal dominating sets equals the family of maximal stable sets. Every minimal dominating or maximal stable set of $\overline{T(2n,n)}$ contains exactly one vertex of each pair of adjacent vertices.

Theorem 4.1.1. *The* Equidomination Function *problem is coNP-complete.*

Proof sketch. A certificate that a function is not an equidominating function consists of two subsets of vertices. Either both subsets are minimal dominating sets but the total weights are distinct. Or they have the same total weight but only one of them is a minimal dominating set. Since all of that can be calculated and checked in polynomial time, the Equidomination Function problem is in *coNP*.

To show hardness one can reduce the so-called Weak Partition problem, which is *NP*-complete (see [60]), to Equidominating Function. For the Weak Partition problem a finite set is given and a positive integer is assigned to each element. It is asked whether there are two disjoint subsets such that the sums of the corresponding integer values of both subsets are equal. If so, the answer is yes. The Weak Partition problem is a special case of the Partition and the Subset Sum problem (see for example [39]).

For a given instance of Weak Partition with n elements, we assign the integer value of each element of Weak Partition as weights to two adjacent vertices of $\overline{T(2n,n)}$. Then the answer to the instance of Weak Partition is yes if and only if the prior defined function is not an equidominating function. □

Furthermore, we can adapt another result of [49]. The proof of Theorem 3.1.3 already indicates that an exponentially large target value can be necessary.

Theorem 4.1.2. *There exist equidominating graphs on* $2n$ ($n \in \mathbb{N}$) *vertices such that the target value* t *of every equidominating structure is of order* $\mathcal{O}\left(\frac{2^n}{\sqrt{n}}\right)$.

Proof sketch. Again, Milanič et al. worked with $\overline{T(2n,n)}$. First, the authors proved that every equistable function on $\overline{T(2n,n)}$ must assign the same weights to adjacent vertices and that the (non-multi) set of weights must have the so-called distinct-subset-sums property. The distinct-subset-sums property requires that the sums of distinct subsets of a set of integers must not be equal. Both holds also for equidominating functions on $\overline{T(2n,n)}$. Lastly, the theorem follows from a result from Erdös [26] that states that the maximum number of a set of n positive integers with the distinct-subset-sum property is of order $\mathcal{O}\left(\frac{2^n}{\sqrt{n}}\right)$. □

The previous two results suggest that any (polynomial time) algorithm for the Equidomination problem would have to use structural properties of equidominating graphs. However, Theorem 3.1.3 shows that this cannot be done with the use of forbidden induced subgraphs. Together, this motivates our belief that the Equidomination problem is at least *NP*-hard and encouraged us to study the Equidomination problem with respect to its parameterized complexity.

4.2 Parameterization

In this section we introduce two parameterized versions of the Equidomination problem. We might call these parameterizations natural parameterizations. A third parameterization, which is a generalization of one of them, will be discussed in Section 5.4.

First of all, we give two definitions that are essential for the parameterized problems and the upcoming investigations.

Definition 4.2.1. *For a given $t \in \mathbb{N}$, a graph G is called **target-t equidominating** if there exists an equidominating structure of the form (ω, t) of G.*

With target-t equidomination we specify equidominating graphs by the additional question which target values can occur. We have already seen, for example, that there is no polynomial $p\colon \mathbb{N} \to \mathbb{N}$ such that $\overline{T(2n,n)}$ is target-$p(n)$ equidominating for all $n \in \mathbb{N}$ (compare Theorem 4.1.2). If a graph is target-t equidominating for some $t \in \mathbb{N}$, there is no direct implication to values less or greater than t. For example, the circle on four vertices C_4 is target-4 equidominating. An equidominating structure is given by (ω, t) with $\omega \equiv 2$ and $t = 4$. But C_4 is neither target-3 nor target-5 equidominating since every minimal dominating set consists of two arbitrary vertices. Thus, t must be even.

By the next definition only weights up to a certain value are allowed.

Definition 4.2.2. *For a given $k \in \mathbb{N}$, a graph G is called **k-equidominating** if there exists an equidominating structure (ω, t) with $\omega\colon V \to [k]$ for some $t \in \mathbb{N}$. In this case, (ω, t) is said to be a **k-equidominating structure** and ω a **k-equidominating function**.*

The significant difference to the definition of equidominating is that the codomain now is $[k]$ instead of $\mathbb{N}$. If a graph is k-equidominating, then it is clearly also k'-equidominating for every $k' \geq k$.

It is easy to see that every target-t equidominating graph is also t-equidominating: every vertex is contained in some minimal dominating set (possibly without any other vertex). Thus, its weight, with respect to any equidominating function, cannot exceed t. The opposite, however, is not true. Consider, for example, for a given $t \in \mathbb{N}$ the edgeless graph $\overline{K_{t+1}}$ on $t+1$ vertices. This graph is not target-t equidominating. But it is 1-equidominating and hence k-equidominating for every $k \in \mathbb{N}$.

Yet, there is an implication from k-equidomination to target-t equidomination. For $k \in \mathbb{N}$, let $G = (V, E)$ be a k-equidominating graph with $|V| = n$ and ω a k-equidominating function. The weight (with respect to ω) of every vertex is at most k. Further, V is trivially a dominating set and thus contains a minimal dominating set. It follows that G has a minimal dominating set of total weight at most kn. This means G is target-t equidominating for some $t \leq k|V|$. We can also consider the total weight of V, which yields $t \leq \omega(V)$ and may sharpen the bound. In reverse, if a graph on n vertices is not target-t equidominating for all $t \leq kn$ ($k \in \mathbb{N}$), then it is also not k-equidominating.

We want to emphasize that most of the results of Section 3.2 do not hold for target-t equidominating and k-equidominating graphs. In their proofs, we often multiply the weight functions and the target values by natural numbers as needed (compare Observation 3.2.1). Thus, we obtain higher values than possibly allowed.

The discussion of Section 4.1 persuades us to study parameterized complexity of the Equidomination problem. To this end, we introduce the following two parameterized versions of the Equidomination problem corresponding to previously stated definitions:

Target-t Equidomination:

Instance: A graph G and $t \in \mathbb{N}$.
Parameter: t.
Problem: Decide whether G is target-t equidominating.

k-Equidomination:

Instance: A graph G and $k \in \mathbb{N}$.
Parameter: k.
Problem: Decide whether G is k-equidominating.

In the next chapter we prove that both the k-equidomination and the Target-t Equidomination problem are fixed-parameter tractable. We do this using the kernelization technique, which we explain in Subsection 2.2.2. For that, we want to find a partition of the vertices of a graph such that two vertices of different blocks of the partition cannot have the same weight with respect to any equidominating function.

If we can find a way to partition the vertices in such a manner, then in some cases one can easily decide that a graph is not target-t equidominating or k-equidominating. We know for both problems how many distinct weights can be allocated. Thus, a graph is not target-t equidominating or k-equidominating if it has more than t or k blocks, respectively.

If we further – in the case of fewer blocks – can somehow bound the size of each block of the partition (in terms of the parameter), we directly get an instance kernel of bounded size: the maximal number of blocks multiplied by the maximal size of each block. More precisely, we cannot find general bounds for the size of each block. However, we can reduce them in such a way that the reduced graph is target-t equidominating or k-equidominating if and only if the original graph is.

The first step is examined in the following Section 4.3 and the second one in Section 5.1. In Section 4.4 we use the obtained partition to develop an *XP* algorithm which can be used for both parameterized problems. In contrast to the *FPT* algorithm, we do not need blocks of bounded size for the *XP* algorithm.

4.3 Decomposition

In this section we derive a partition of the vertices of a graph that meets the condition that different blocks must have different weights regarding an equidominating structure (see Corollary 4.3.18). We start with examining the twin partition with respect to equidomination, which we then coarsen to obtain the so-called pseudo class partition.

4.3.1 Twin Partition

The first step to obtain the desired partition is the introduction of twins. As we will see, it provides a helpful frame when dealing with equidomination. It is also used in the context of equistability (see for example [38]).

Definition 4.3.1. *Let $G = (V, E)$ be a graph. Two vertices v, $w \in V$ are called **twins** if they have the same neighborhood except themselves, that is if*

$$N(v) \setminus \{w\} = N(w) \setminus \{v\} .$$

*If further $vw \in E$, we say v and w are **true twins** and otherwise **false twins**.*

We define a relation using the twin property:

$$v \sim_t w :\Longleftrightarrow N(v) \setminus \{w\} = N(w) \setminus \{v\} .$$

*This relation will be referred to as the **twin relation**.*

Since $\sim_t$-related vertices dominate the same set of vertices (besides themselves in the case of false twins), it seems reasonable to work with the twin relation in the context of (minimal) dominating sets.

Lemma 4.3.2. *The twin relation is an equivalence relation.*

Proof. Let G be a graph with vertex set V and edge set E. Symmetry and reflexivity follow immediately from the definition. For transitivity, let v, w and $x \in V$ with $v \sim_t w$ and $w \sim_t x$. Then $N(v) \setminus \{w, x\} = N(w) \setminus \{v, x\} = N(x) \setminus \{v, w\}$ and for symmetry reasons it remains to show that $vw \in E$ if $wx \in E$. So let $wx \in E$. Since v and w are related, we get $vw \in E$ and the proof is finished. □

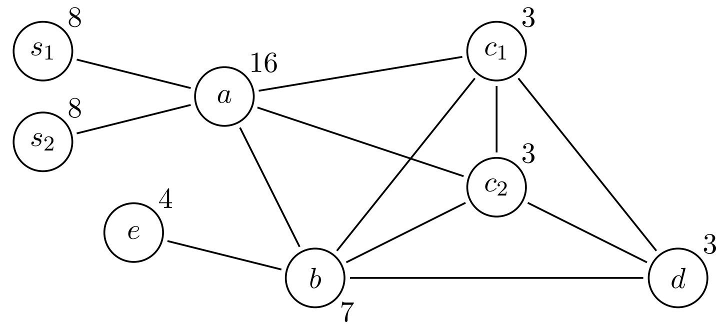


Figure 4.1: An equidominating graph on 8 vertices; the target value is $t = 23$; its twin partition is $\{\{a\}, \{b\}, \{c_1, c_2\}, \{d\}, \{e\}, \{s_1, s_2\}\}$ and its pseudo class partition $\{\{a\}, \{b\}, \{c_1, c_2, d\}, \{e\}, \{s_1, s_2\}\}$.

Since the considered partition is based on an equivalence relation – more precisely on equivalence classes – we will use the term class instead of blocks in the following. The equivalence classes of $\sim_t$ are called **twin classes** and the partition of the vertices into twin classes is called **twin partition**. Similar to the proof of Lemma 4.3.2 one can show that all vertices of a twin class are either pairwise adjacent or pairwise non-adjacent. Therefore, twin classes are specified to be **clique classes** in the first and **stable set classes** in the latter case.

A twin class can also be a single vertex. Even though a single vertex is strictly speaking a stable set as well as a clique, we use the terms clique class and stable set class only for twin classes with at least two elements. We call a twin class with one vertex a **singleton class**.

In the following, we refer several times to the graph considered in Chapter 3. To the reader's convenience, we state it here again in Figure 4.1. Note that we slightly varied the equidominating structure. The vertices c_1 and c_2 form a clique class, the vertices s_1 and s_2 form a stable set class, and all other vertices are singleton classes.

Now, let T_1 and T_2 be two twin classes. It is easy to see that either every vertex of T_1 is adjacent to every vertex of T_2 or every vertex of T_1 is non-adjacent to every vertex of T_2. In the first case we say that T_1 and T_2 **see** each other and that T_1 sees T_2 and vice versa. We also say that a vertex and a twin class see each other, and likewise two vertices. Furthermore, if appropriate, we use expressions for twin classes which are usually used for vertices (for example, a twin class is adjacent to, dominates, is dominated by, et cetera).

For the sake of completeness, we also define the **quotient graph** $Q(G)$ of a graph G: every twin class of G is a vertex of $Q(G)$ and two vertices are adjacent if and only if the corresponding twin classes see each other. A twin class is a special case of a module and hence the decomposition into twin classes is a special form of a modular decomposition. Therefore, the twin partition can be computed in linear time using one of the modular decomposition algorithms of [22], [47] and [59].

As the following observations show, the twin relation is a helpful instrument with regard to minimal dominating sets and thus also with regard to equidomination.

Observation 4.3.3. *For every minimal dominating set D and every stable set class S, we have $|D \cap S| \in \{0, 1, |S|\}$.*

This observation indicates that we have to look carefully at stable set classes with two elements. If a stable set class S contains only two vertices, then Observation 4.3.3 includes all possible cases. In particular, it can not occur that more than one but not all vertices of S are elements of a minimal dominating set. It turns out that stable set classes of size two indeed play a special role (see Lemma 4.3.13 and Definition 4.3.14).

Since, in contrast to stable set classes, the vertices of a clique class do dominate each other, the situation is less complicated here.

Observation 4.3.4. *For every minimal dominating set D and every clique class C, we have $|D \cap C| \in \{0, 1\}$.*

The next observation holds not only for stable set classes but also for stable sets in general. It is based on the fact that every maximal stable set is also a minimal dominating set. Therefore, every stable set is contained in at least one minimal dominating set.

Observation 4.3.5. *For every stable set S and for every equidominating structure (ω, t) it holds that $\omega(S) \leq t$ and hence $|S| \leq t$.*

Unfortunately, vertices of different twin classes of an equidominating graph can have the same weight in an equidominating structure. This means that the twin partition does not meet the desired condition discussed at the end of the previous section. For example, let us take a look again at the equidominating structure of the graph of Figure 4.1. The vertices c_1 and d do not lie in the same twin class since only c_1 is adjacent to a. However, they both have the same weight. Therefore, we have to find another way to partition the vertices which needed a careful and intense analysis.

We want to mention that this is one of the significant differences between equidomination and equistability. With respect to an equistable function, two vertices of different twin classes always have different weights [43].

4.3.2 Pseudo Class Partition

Motivated by the previous discussion, we now examine in which cases vertices of different twin classes of an equidominating graph can have equal weights. It turns out that two vertices can only have the same weight if they lie in the same twin class or are adjacent. That means on the one hand that when trying to construct an equidominating structure one has to consider fewer combinatorial possibilities. And on the other hand, that for a given number of weights to be allocated one can bound the diameter of an equidominating graph.

Eventually, this examination leads us to a partition that coarsens the twin partition. To start with, we introduce the following term.

Definition 4.3.6. *Let $G = (V, E)$ be a graph. Two vertices x, $y \in V$ are called* ***mds-exchangeable*** *if and only if*

(i) there exists a minimal dominating set $D \subseteq V$ with $|\{x, y\} \cap D| = 1$, and

(ii) for all minimal dominating sets $D \subseteq V$ with $|\{x, y\} \cap D| = 1$ the symmetric difference $(D \setminus \{x, y\}) \cup (\{x, y\} \setminus D)$ is a minimal dominating set.

Loosely speaking, two vertices are mds-exchangeable if they can be exchanged for each other in any minimal dominating set containing exactly one of them. Clearly, if two vertices of an equidominating graph are mds-exchangeable, then they must have the same weight in every equidominating function. To ensure this implication, we require in the definition that there is a minimal dominating set that contains only one of the vertices. Indeed, this implication initially motivated the introduction of mds-exchangeability.

Actually, for nearly all pairs of vertices there exists a minimal dominating set that contains exactly one of the two vertices. The only exceptional case are false twins of a stable set class. For such two vertices it can occur that either none or both vertices are in a minimal dominating set. For example, for this reason the vertices s_1 and s_2 of the graph in Figure 4.1 are not mds-exchangeable. Furthermore, c_1, c_2 and d of this graph are pairwise mds-exchangeable.

We remark that it can occur that two mds-exchangeable vertices both are elements of one minimal dominating set. For example, all vertices of C_4 are pairwise mds-exchangeable while every minimal dominating set of C_4 contains exactly two arbitrary vertices.

It is easy to see that two vertices of a clique class are mds-exchangeable. Thus, we get the following observation.

Observation 4.3.7. *Every equidominating function is constant on each clique class.*

Analogously, if the first part of Definition 4.3.6 is fulfilled, then the same holds for stable set classes.

Observation 4.3.8. *If there exists a minimal dominating set containing exactly one vertex of a stable set class, then every equidominating function is constant on that stable set class.*

As already mentioned, every (non-maximal) stable set can be extended to a minimal dominating set. Such an extension of a stable set to a minimal dominating set can be done in different ways. One way is the following: for a given stable set $S \subseteq V$ of a graph

$G = (V, E)$ we determine a minimal dominating set D' of $G[V \setminus N[S]]$. We claim that the disjoint union $D := S \dot{\cup} D'$ is a minimal dominating set of G. The domination property of D is obvious. Every vertex of S is its own private neighbor (possible among others). Furthermore, $\emptyset \neq pn(v, D') = pn(v, D)$ holds for every vertex $v \in D'$. That means it is not possible to delete any vertex from D and remain dominating in G. Hence, we obtain the following observation, which we often use in the proofs of the upcoming lemmas.

Observation 4.3.9. *For every stable set S there exists a minimal dominating set D with $S \subseteq D$ and $D \cap N(S) = \emptyset$.*

After these preliminaries, we show in which cases vertices of different twin classes cannot have the same weight. We remark that whenever we speak about weights, we (implicitly) require the considered graph to be equidominating.

Lemma 4.3.10. *Let $G = (V, E)$ be an equidominating graph with equidominating structure (ω, t) and let x, $y \in V$ be two vertices of different twin classes with* $\operatorname{dist}(x, y) \geq 2$. *Then $\omega(x) \neq \omega(y)$ holds.*

Proof. Suppose there are two such vertices x, $y \in V$ with $\omega(x) = \omega(y)$. First, let one of the vertices be in a clique class C, say $x \in C$. Extend $\{x, y\}$ to a minimal dominating set and then exchange another vertex of C for y. We get a subset of total weight t, by Observation 4.3.7, which is not a minimal dominating set, by Observation 4.3.4, a contradiction.

So, let both x and y be either elements of singleton classes or stable set classes. As they are in different classes, without loss of generality let v be a vertex seen by x and not seen by y. By Observation 4.3.9, we can extend $\{y, v\}$ to a minimal dominating set D such that $D \cap N(\{y, v\}) = \emptyset$. If we exchange y for x, again we get a subset (namely $D - y + x$) of total weight t, that is not a minimal dominating set since y is no longer dominated, a contradiction. □

Note that in the previous lemma the two mentioned vertices must be of two different twin classes. Two elements of a stable set class, of course, can have the same weight while always having distance at least two.

In the following, we take a closer look at adjacent vertices, where we find a slightly more complicated situation. We begin by showing that vertices of adjacent stable set classes and clique classes cannot have the same weight.

Lemma 4.3.11. *Let $G = (V, E)$ be an equidominating graph with equidominating structure (ω, t) and let $S \subseteq V$ be a stable set class and $C \subseteq V$ a clique class that see each other. Then $\omega(x) \neq \omega(y)$ holds for all $x \in S$ and for all $y \in C$.*

Proof. Suppose there are vertices $x \in S$ and $y \in C$ with $\omega(x) = \omega(y)$. The other vertices of S must have a different weight than x and y. Otherwise, one could extend

S to a minimal dominating set D such that $D \cap N(S) = \emptyset$ (see Observation 4.3.9) and exchange two vertices of S for two vertices of C while maintaining the total weight t. This contradicts Observation 4.3.4.

So, extend S to a minimal dominating set D such that $D \cap N(S) = \emptyset$. The set $D' := D - x + y$ has weight t and thus is also a minimal dominating set. Now $D' - x' + x$ ($x' \in S$, $x' \neq x$) has a weight different than t, but since the swapped vertices lie in the same twin class, it still must be a minimal dominating set, again a contradiction. □

Further, with respect to any equidominating function adjacent vertices of a stable set class and a singleton class cannot have the same weight.

Lemma 4.3.12. *Let $G = (V, E)$ be an equidominating graph with equidominating structure (ω, t). Let $S \subseteq V$ be a stable set class and let $\{y\}$ be a singleton class with $y \in N(S)$. Then $\omega(x) \neq \omega(y)$ holds for all $x \in S$.*

Proof. Let x, $x' \in S$ and $y \in N(S)$ be an adjacent singleton class. Suppose that $\omega(x) = \omega(y)$. The set $N(S) \setminus N[y]$ cannot be empty, since otherwise one could extend $\{y\}$ to a minimal dominating set D such that $D \cap N(y) = \emptyset$ (see Observation 4.3.9). But now $D - y + x$ does not dominate x', which contradicts $\omega(x) = \omega(y)$. Furthermore, there must be a vertex $v \in N(S) \setminus N[y]$ with $N[v] \subseteq N[S]$ (otherwise $\{y\} \cup (V \setminus N[S])$ is a dominating set, which contains a minimal dominating set D with $y \in D$ and $x \notin D$, but $D - y + x$ does not dominate x').

So, extend $\{v, y\}$ to a minimal dominating set D such that $D \cap N(\{v, y\}) = \emptyset$. Since $D' := D - y + x$ has total weight t and thus is a minimal dominating set, $pn(v, D') = S \setminus \{x\}$. It follows that $D' - v + y$ is also a minimal dominating set, which implies $\omega(v) = \omega(y)$, a contradiction to Lemma 4.3.10. □

The proof of Lemma 4.3.12 suggests that if v and y lay in one twin class, x and y could have the same weight. As the next lemma shows, this is indeed possible, but only in a specific situation.

Lemma 4.3.13. *Let $G = (V, E)$ be an equidominating graph with equidominating structure (ω, t) and let S_1, $S_2 \subseteq V$ be two adjacent stable set classes. Further, let $x \in S_1$, $y \in S_2$ be two vertices with $\omega(x) = \omega(y)$. Then the following assertions hold:*

(i) $|S_1| = |S_2| = 2$,

(ii) ω *is constant on* $S_1 \cup S_2$,

(iii) *every twin class seen by* S_1 *is also seen by* S_2 *and vice versa.*

Furthermore, if two adjacent stable set classes of size two have the same closed neighborhood, then all vertices of those stable set classes have the same weight in any equidominating structure.

Proof. Suppose one of the stable set classes has more than two elements, say S_1. We extend S_1 to a minimal dominating set D with $D \cap S_2 = \emptyset$. Then, $D - x + y$ has total weight t, but following Observation 4.3.3 it is not a minimal dominating set. So, assertion (i) is shown and thus let $S_1 = \{x, x'\}$ and $S_2 = \{y, y'\}$.

Let D^* be a minimal dominating set of $G[V \setminus N[S_1 \cup S_2]]$. Then, $D \coloneqq D^* \cup \{x, y\}$ is a minimal dominating set of G (note that $y' \in pn(x, D)$ and $x' \in pn(y, D)$). Observation 4.3.8 yields assertion (ii).

Now, suppose S_2 sees a vertex v which is not seen by S_1. Extend $S_1 \cup \{v\}$ to a minimal dominating set D^* such that $D^* \cap S_2 = \emptyset$. Then $D \coloneqq (D^* \setminus S_1) \cup S_2$ has total weight t, but D is not a minimal dominating set, as $N[D] = N[D \setminus \{y\}]$. That proves assertion (iii).

The last statement of the lemma follows if we can show that the vertices of such stable set classes are pairwise mds-exchangeable. For that, let S_1 and S_2 be two adjacent stable set classes of size two with $N[S_1] = N[S_2]$ and let x, $y \in S_1 \cup S_2$. If x and y are adjacent, it is clear that there are minimal dominating sets containing either x or y. Further, we have already seen above that there exists a minimal dominating set containing only one vertex if x and y are in the same stable set class.

Now, let D be a minimal dominating set with $x \in D$ and $y \notin D$. It holds that $1 \leq |D \cap (S_1 \cup S_2)| \leq 2$. In both cases we get $pn(y, D - x + y) = pn(x, D)$. Since x and y see the same vertices in $V \setminus S_1 \cup S_2$, $pn(v, D) = pn(v, D - x + y)$ for all $v \in D - x$. Hence, $D - x + y$ is a minimal dominating set and the proof is finished. □

As a consequence of Lemma 4.3.13, there can be an arbitrarily large number of stable set classes of size two with vertices of the same weight in an equidominating graph. Such an occurrence could be a problem when trying to achieve bounded kernels for the parameterized problems. But the good thing is that all of those stable set classes both see each other and see the same twin classes in the remainder of the graph. Therefore, as we will see in Subsection 5.1, it is possible to reduce them to a manageable number.

For a better handling, we introduce the following new term.

Definition 4.3.14. *Let $G = (V, E)$ be a graph and $\mathcal{S} \subseteq V$ be a maximal subset such that:*

(i) $G[\mathcal{S}] \cong T(2n, n)$ for some $n \geq 2$,

(ii) $\mathcal{S}$ is a module.

Then $\mathcal{S}$ is called a ***stable set bundle****.*

Here, maximal means that no other subset fulfills the two conditions and properly contains $\mathcal{S}$. Every stable set bundle contains several stable set classes of size two with the same neighborhood outside the stable set bundle. Stable set bundles behave similar to clique classes. In fact, a stable set bundle forms a clique class in the quotient graph. Considered the other way around, adding a false twin to every vertex of a clique class

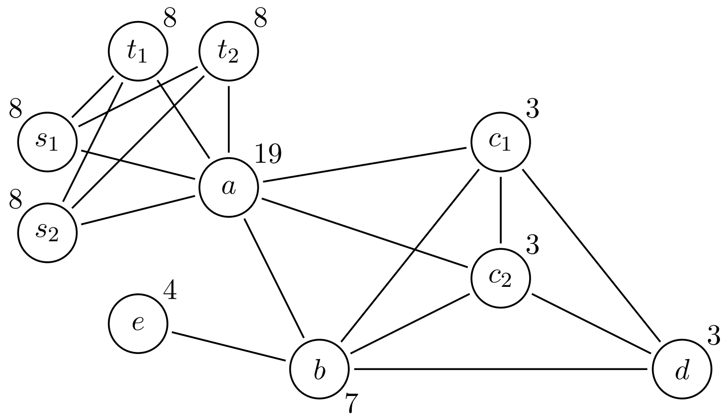


Figure 4.2: An equidominating graph on 10 vertices; the target value is $t = 23$; its twin partition is $\{\{a\},\{b\},\{c_1,c_2\},\{d\},\{e\},\{s_1,s_2\}\},\{t_1,t_2\}\}$ and its pseudo class partition $\{\{a\},\{b\},\{c_1,c_2,d\},\{e\},\{s_1,s_2,t_1,t_2\}\}$.

yields a stable set bundle. Following Lemma 4.3.13, the vertices of a stable set bundle are pairwise mds-exchangeable and, therefore, every equidominating function is constant on a stable set bundle. In the graph of Figure 4.2 the four vertices s_1, s_2, t_1 and t_2 form a stable set bundle.

Now, regarding whether two vertices can have the same weight in an equidominating structure, the last open question is: can vertices of a clique class or a singleton class have the same weight as its neighboring clique class or singleton class? The answer to this question is yes: there can be a clique, that is not a clique class, whose vertices are pairwise mds-exchangeable.

Definition 4.3.15. *Let G be a graph and $\mathcal{C}$ an inclusion-wise maximal clique of pairwise mds-exchangeable vertices that contains at least two twin classes. Then $\mathcal{C}$ is called a* ***clique bundle****.*

Upon first reading it seems a little bit odd to define clique bundles exactly as what we are looking for: pairwise mds-exchangeable vertices of possibly different twin classes. However, the crucial thing here is that we can identify clique bundles efficiently (see Algorithm 1 below).

In a clique bundle there can be both clique classes and singleton classes but clearly no stable set classes. If it is not relevant whether we talk about a clique class or a singleton class, then we simply refer to a twin class of a clique bundle. We require at least two twin classes to be in a clique-bundle in order that a (single) twin class is not a clique bundle and a twin class at the same time. This is needed later on (compare Corollary 4.3.17).

In the graph shown in Figure 4.2 the clique class $\{c_1, c_2\}$ and the singleton class $\{e\}$ together form a clique bundle. Further, one can see an equidominating graph which

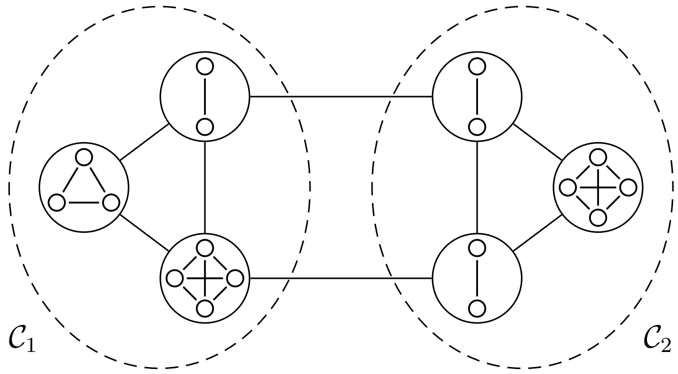


Figure 4.3: An equidominating graph consisting of the two clique bundles $\mathcal{C}_1$ and $\mathcal{C}_2$ each containing three clique classes with two to four vertices; clique classes are indicated by circles and an edge between the circles of two clique classes represents all edges between the vertices of the corresponding clique classes.

consists of two clique bundles in Figure 4.3. In this graph, every minimal dominating set contains exactly one vertex of each clique bundle.

We use the term **bundle** to refer to either a stable set bundle or a clique bundle. As a bundle consists of several twin classes, we use calligraphic style variables for bundles. Slightly abusing notation, we will also refer to the twin classes of a bundle and write $T \in \mathcal{B}$, for a twin class T and a bundle $\mathcal{B}$.

Recall that a twin class is either a stable set class, a clique class or a singleton class. Due to the existence of bundles we introduce a coarsening of twin classes:

Definition 4.3.16. *A **pseudo class** is either a twin class not contained in a bundle or a stable set bundle or a clique bundle.*

That is, a pseudo class is exactly one of following: a) a singleton class, b) a stable set class, c) a clique class, d) a stable set bundle or e) a clique bundle. For example, the stable set classes $\{s_1, s_2\}$ and $\{t_1, t_2\}$ of the graph shown in Figure 4.2 are no pseudo classes. However, taken together they form a pseudo class which is a stable set bundle. By this definition and the previous discussion, we get the following result.

Corollary 4.3.17. *There is a unique partition of the vertices of a graph into pseudo classes.*

We will refer to this unique partition as the **pseudo class partition**. The pseudo class partition of the graph of Figure 4.1 is $\big\{\{a\}, \{b\}, \{c_1, c_2, d\}, \{e\}, \{s_1, s_2\}\big\}$ and the one of Figure 4.2 is $\big\{\{a\}, \{b\}, \{c_1, c_2, d\}, \{e\}, \{s_1, s_2, t_1, t_2\}\big\}$.

The following corollary is a summary of the previous lemmas. It states that the pseudo class partition fulfills the condition that the vertices of different blocks of the partition cannot have the same weight with respect to any equidominating structure.

Corollary 4.3.18. *Let G be an equidominating graph with equidominating structure (ω, t) and P_1, P_2 be two different pseudo classes. Then $\omega(x) \neq \omega(y)$ holds for all $x \in P_1$ and $y \in P_2$.*

Furthermore, using the pseudo class partition, we get the following lemma about the relation of two subsets of vertices regarding domination. It also holds for the twin partition which can be proved analogously.

Lemma 4.3.19. *Let $G = (V, E)$ be a graph with pseudo class partition $\{P_1, \ldots, P_s\}$. Let D_1, $D_2 \subseteq V$ be two subset of vertices with $|D_1 \cap P_i| = |D_2 \cap P_i|$ for all $i = 1, \ldots, s$. Then D_1 is a minimal dominating set if and only if D_2 is a minimal dominating set.*

Proof. Due to symmetry reasons, it is sufficient to show that D_2 is a minimal dominating set if D_1 is a minimal dominating set. Let D_1 be a minimal dominating set. On singleton classes D_1 and D_2 clearly are identical. Since the vertices of a clique class, a clique bundle and a stable set bundle are pairwise mds-exchangeable, we may assume that $D_1 \cap P_i = D_2 \cap P_i$ for all pseudo classes P_i that are not stable set classes, $i = 1, \ldots, s$.

For any stable set class $S \in \{P_1, \ldots, P_s\}$, we have $|D_1 \cap S| \in \{0, 1, |S|\}$, by Observation 4.3.3. So $D_1 \cap S \neq D_2 \cap S$ only if $|D_1 \cap S| = 1$. However, in this case the vertices of the stable set class S are pairwise mds-exchangeable. That means we can exchange the vertices of $D_1 \cap S$ with vertices of $D_2 \cap S$ without losing the property of being a minimal dominating set. □

This means, regarding the question whether a subset of vertices is a minimal dominating set, it does not matter which vertices of a pseudo class are in the subset. Only the number of vertices counts. A fact we use to construct an *XP* algorithm for the k-equidominating problem in Section 4.4.

What is missing is a possibility to determine the pseudo class partition. For that, we first compute the twin partition. As already mentioned, there are several linear time algorithms for this computation (see [22], [47] and [59]). In some cases twin classes are certainly also pseudo classes, for example, stable set classes with more than two vertices. However, for singleton classes, clique classes and stable set classes of size two we need a way to decide whether they form a pseudo class on their own or whether they lie in a bundle. This means concretely that we have to decide whether two adjacent vertices are mds-exchangeable.

For that we developed Algorithm 1. For two vertices v_1 and v_2, it checks if there is a private neighbor v' of v_1 in any dominating set not containing v_2, that is not seen by v_2. If so, the two vertices are not mds-exchangeable. See Figure 4.4 for a better understanding of Algorithm 1.

Theorem 4.3.20. *Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = M$ and let x, $y \in V$. Algorithm 1 correctly decides whether x and y are mds-exchangeable and runs in $\mathcal{O}(nm)$ time.*

Algorithm 1 Checking adjacent vertices for mds-exchangeability

Input: Two adjacent vertices $x, y \in V$ of a graph $G = (V, E)$
Output: **YES**, if x and y are mds-exchangeable, otherwise **NO**

1: **for all** $(v_1, v_2) \in \{(x, y), (y, x)\}$ **do** ▷ Check both combinations
2:     **for all** $v' \in N(v_1) \setminus N[v_2]$ **do**
3:         **if** $\{v_1\} \cup \Big(V(G) \setminus \big(N[v'] \cup \{v_2\}\big)\Big)$ is a dominating set **then**
4:             return **NO** ▷ v' is private neighbor of x
5:         **end if**
6:     **end for**
7: **end for**
8: return **YES** ▷ x and y are mds-exchangeable

Proof. First we prove the correctness. Since the algorithm checks both combinations (see line 1), the following argumentation holds for $v_1 = x$, $v_2 = y$ as well as for $v_1 = y$, $v_2 = x$. If the output is NO, the stated set in line 3 contains a minimal dominating set D with $v' \in pn(v_1, D)$ and $v' \notin N[v_2]$. So, v' is not dominated by $D - v_1 + v_2$ and thus x and y cannot be mds-exchangeable. If the output is YES and the algorithm reaches line 8, there is no vertex v' in $N(v_1)$, which is not adjacent (or equal) to v_2 and which is a private neighbor of v_1 in any minimal dominating set that does not contain v_2. So, we cannot find a minimal dominating set D with $v_1 \in D$, $v_2 \notin D$ and $pn(v_1, D) \cap N[v_2] \neq \emptyset$. Further, since v_1 and v_2 are adjacent, there are minimal dominating sets that contain only v_1 and that contain only v_2. Taken together, x and y are mds-exchangeable.

It is clear that the algorithm terminates for a finite graph. There are $\mathcal{O}(n)$ vertices in $N(v_1) \setminus N[v_2]$. For each vertex $v' \in N(v_1) \setminus N[v_2]$ it is sufficient to check if the vertices of $N(v') \setminus \{v_1\}$ are dominated since every other vertex is an element of the set stated in line 3. With the use of two flags for each vertex and a global counter we can do this considering every edge two times. So, line 3 needs $\mathcal{O}(m)$ time which leads to a total running time of $\mathcal{O}(nm)$. □

After computing the twin partition, one can apply Algorithm 1 to adjacent clique classes and singleton classes, and adjacent stable set classes of size two to find all clique bundles and stable set bundles, respectively. Of course, one could also check the neighborhoods of adjacent stable set classes of size two to discover stable set bundles.

To discover all bundles one has to apply Algorithm 1 for every edge, which gives us a total running time of $\mathcal{O}(nm^2)$. Hence, we get the following corollary.

Corollary 4.3.21. *The pseudo class partition of a graph with n vertices and m edges can be computed in time $\mathcal{O}(nm^2)$.*

We want to close this section with a brief discussion about different ways to partition the vertices of a graph in the context of the results of this section. One might ask why

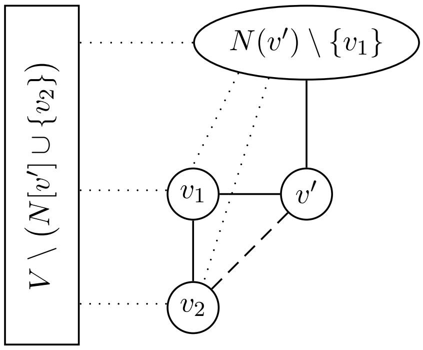


Figure 4.4: A visualization of the idea of Algorithm 1; the dashed edge is not existent and the dotted edges symbolize edges that may exist. If the set $\{v_1\} \cup \Big(V \setminus \big(N[v'] \cup \{v_2\}\big)\Big)$ is a dominating set (compare line 3), then v' is a private neighbor of v_1 in some minimal dominating set and since $v_2v' \notin E$, v_1 and v_2 are not mds-exchangeable.

we do not simply use the property of being mds-exchangeable to partition the vertices. Indeed, mds-exchangeability is an equivalence relation and thus the equivalence classes yield a partition.

Let us call the equivalence classes of the mds-exchangeable relation **mds-ex classes** and the corresponding partition the **mds-ex partition**. The good thing about using the mds-ex partition is that vertices of the same mds-ex class clearly must have the same weight in every equidominating structure – this is what motivated us to introduce mds-exchangeability.

However, the mds-ex partition fails the desired condition that vertices of different classes must have different weights with respect to any equidominating function. As already discussed, two vertices of a stable set class may not be mds-exchangeable because there exists no minimal dominating set containing only one of them. But such two vertices could have the same weight (see for example s_1 and s_2 in Figure 4.1).

Moreover, it would be nice to gain independence from equidomination for the results of this section (in the previous lemmas we always require the graphs to be equidominating). This is indeed possible for Lemma 4.3.10, Lemma 4.3.11 and Lemma 4.3.13. In their proofs, we exchange a vertex of a minimal dominating set with another vertex and show that the resulting set is not a minimal dominating set. Consequently, the two vertices are not mds-exchangeable. Therefore, we get the following corollaries, which are non-equidominating variants of the three lemmas.

Corollary 4.3.22. *Let $G = (V, E)$ be a graph and $x, y \in V$ be two vertices of different twin classes with* $\operatorname{dist}(x, y) \geq 2$. *Then x and y are not mds-exchangeable.*

Corollary 4.3.23. *Let $G = (V, E)$ be a graph and let $S \subseteq V$ be a stable set class and $C \subseteq V$ a clique class that see each other. Then x and y are not mds-exchangeable for* *all $x \in S$ and for all $y \in C$.*

Corollary 4.3.24. *Let $G = (V, E)$ be a graph and let S_1, $S_2 \subseteq V$ be two adjacent stable set classes. Further let $x \in S_1$, $y \in S_2$ be two mds-exchangeable vertices. Then the following assertions hold:*

(i) $|S_1| = |S_2| = 2$,

(ii) the vertices of $S_1 \cup S_2$ are pairwise mds-exchangeable ,

(iii) every twin class seen by S_1 is also seen by S_2 and vice versa .

Furthermore, if two adjacent stable set classes of size two have the same closed neighborhood, then all vertices of those stable set classes are pairwise mds-exchangeable.

However, Lemma 4.3.12 is a slightly different. In its proof, we use that if two vertices of an equidominating graph can be exchanged in one minimal dominating set (such that the resulting set is a minimal dominating set), then this is also the case in every other minimal dominating set. Thus, the two vertices are mds-exchangeable and have the same weight in every equidominating structure.

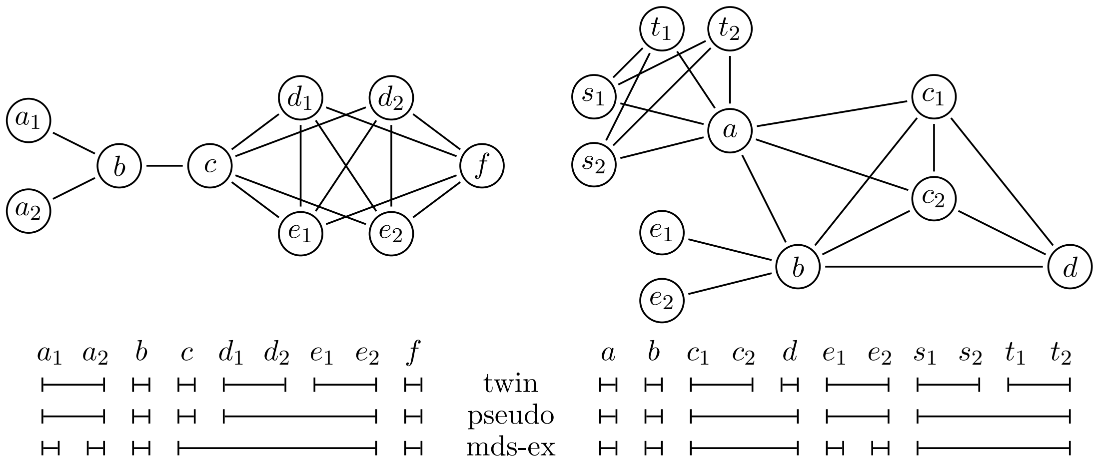
However, this holds not in general. For example, let us take a look at the path P_6 on the vertices $v_1, \ldots, v_6$ with $v_i v_j \in E(P_6)$ if and only if $|i - j| = 1$ $(1 \leq i < j \leq 6)$. We can exchange v_1 for v_2 in the minimal dominating set $\{v_2, v_4, v_6\}$ and the resulting set $\{v_1, v_4, v_6\}$ is a minimal dominating set, too. This is not the case for the minimal dominating set $\{v_2, v_5\}$.

This means that there is no non-equidominating version of Lemma 4.3.12. Indeed, the vertices c (singleton class) and d_1 (element of a stable set class) of the graph in Figure 4.5(a) are mds-exchangeable. Furthermore, one can see in Figure 4.5 a comparison between the twin partition, the pseudo class partition and the mds-ex partition: on the left side for a non-equidominating graph and on the right side for the yet considered, once more extended equidominating graph.

Regarding the relation of the different partitions, the following holds:

- For any graph, the pseudo class partition is a coarsening of the twin partition.
- For an equidominating graph, the mds-ex partition is a refinement of the pseudo class partition.

It seems that the mds-ex partition is not very helpful with respect to our investigations. However, there are at least two reasons for considering the mds-ex partition. First, we can use it to decide that a graph is not equidominating. This is the case if the mds-ex partition does not refine the pseudo class partition. Secondly, the mds-ex partition tells us which vertices definitely must have the same weight in any potential equidominating structure. Both can be used for an implementation of the *XP* algorithm for the k-Equidomination problem presented in the next section.



(a) non-equidominating graph **(b)** equidominating graph

Figure 4.5: A comparison of the twin partition, the pseudo class partition and the mds-ex partition; the lines under the vertices show which vertices lie in the same class of the respective partition.

We can compute the mds-ex partition similar to the pseudo class partition. However, we need a possibility to decide whether the vertices of a stable set class S are mds-exchangeable or not. More precisely, we have to check if there exists a minimal dominating set that contains exactly one vertex of S.

For that, we can proceed as follows. Let $G = (V, E)$ be a graph, $S \subseteq V$ a stable set class and $s \in S$. For each $x \in N(S)$, we check if $\{s\} \cup (V \setminus N[x])$ is a dominating set (we remark that $S \subseteq N[x]$). If this is the case for one $x \in N(S)$, then there is a minimal dominating set $D \subseteq \{s\} \cup (V \setminus N[x])$ with $x \in pn(s, D)$ and $S \cap D = 1$. Hence, the vertices of S are pairwise mds-exchangeable. Otherwise, the vertices of S are not mds-exchangeable.

4.4 XP Algorithm

In this section we describe an *XP* algorithm which decides whether a given graph is k-equidominating for some fixed $k \in \mathbb{N}$. In the running time of this algorithm only k appears in the exponents but not the size of the graph (compare Subsection 2.2.2). The aim is to apply this algorithm to the constructed kernels of the parameterized problems to achieve *FPT* algorithms. The algorithm mainly follows the ideas and the algorithm for the k-Equistability problem of Levit et al. [36], [43].

The basic idea of the algorithm is that by considering the pseudo class partition one does not have to examine every possible weight function nor every possible subset of vertices. Since different vertices of the same pseudo class, roughly said, play the same role regarding minimal dominating sets (compare Lemma 4.3.19), two weight functions that differ only by switched weights for vertices of the same pseudo class can be handled as the same. This leads to (equivalence) classes of weight functions.

Further, we reduce the running time from a brute force algorithm by classifying subsets of vertices, again using Lemma 4.3.19. As a result, we have to check only one subset per class for being a minimal dominating set.

Algorithm 2 An *XP* algorithm for the k-EQUIDOMINATION problem

```
Input: A graph G = (V, E), k ∈ ℕ
Output: a k-equidominating structure if G is k-equidominating, otherwise NO
determine pseudo classes P_1, ... P_s by computing the twin partition and identifying
  clique bundles and stable set bundles via Algorithm 1
if s > k then
    return NO                                        ▷ G cannot be k-equidominating
end if
compute an arbitrary minimal dominating set D
compute the set Ω of weight functions to check
for all ω ∈ Ω do
    compute t_ω = Σ_{v∈D} ω(v)
    for all x ∈ X_ω do
        compute an arbitrary S ∈ 𝒮_ω(x)
        if S is a minimal dominating set then
            if Σ_{i=1}^{k} i x_i ≠ t_ω then          ▷ there is a minimal dominating
                next ω                                  set of weight unequal to t_ω
            end if
        else if Σ_{i=1}^{k} i x_i = t_ω then         ▷ there is a set of weight t_ω which
            next ω                                      is not a minimal dominating set
        end if
        return (ω, t_ω)                              ▷ ω is a k-equidominating function
    end for
end for
return NO                                ▷ no k-equidominating function was found
```

Theorem 4.4.1. *For a given $k \in \mathbb{N}$, it is decidable whether a graph $G = (V, E)$ is k-equidominating or not in time $\mathcal{O}\left(nm^2 + n^k k^k + n^{2k+2} k^{-k-1} + k^{3k+3}\right)$ (with $|V| = n$ and $|E| = m$). Furthermore, a k-equidominating structure is computed in this time if G is k-equidominating.*

Proof. We first discuss how the algorithm works and compute its running time afterward. There are k^n different weight functions from V to $[k]$ we would have to test and 2^n potential minimal dominating sets. As we will see, we can reduce both numbers using the pseudo class partition.

Let $(P_1, \dots, P_s)$ be the partition of V into pseudo classes. If $s > k$, G is not k-equidominating and we are done. Otherwise, we define an equivalence relation on the set of weight functions as follows:

$$\omega_1 \sim \omega_2 :\Longleftrightarrow \forall i \in [r]\ \forall j \in [k]\colon\ \left|\omega_1^{-1}(j) \cap P_i\right| = \left|\omega_2^{-1}(j) \cap P_i\right| .$$

In simple terms two weight functions are equivalent if they assign each weight to the same number of vertices within every pseudo class.

We claim that either every function of an equivalence class is a k-equidominating function or none. Basically, this is a direct consequence of Lemma 4.3.19. To prove the claim, let $\omega_1, \omega_2\colon V \to [k]$ be two weight functions with $\omega_1 \sim \omega_2$ and let ω_1 be a k-equidominating function. Further, let D be a minimal dominating set. Since $\omega_1 \sim \omega_2$, there is a subset $D' \subseteq V$ with $\omega_1(D') = \omega_2(D)$ and $|D' \cap P_i| = |D \cap P_i|$ for all $i = 1, \dots, s$. Following Lemma 4.3.19, D' is a minimal dominating set, too, and hence $t = \omega_1(D') = \omega_2(D)$.

Now, let $X \subseteq V$ with $\omega_2(X) = t$. We construct $X' \subseteq V$ by exchanging each vertex $v \in X$ with a vertex v' of the same pseudo class as v with $\omega_1(v') = \omega_2(v)$. This is possible since $\omega_1 \sim \omega_2$. We get $\omega_1(X') = t$ and $|X' \cap P_i| = |X \cap P_i|$ for all $i = 1, \dots, s$. Since ω_1 is a k-equidominating function, X' is a minimal dominating set and again, by Lemma 4.3.19, X is a minimal dominating set, too.

Consequently, instead of checking every possible weight function, it is sufficient to check just one representative of each equivalence class. Further, by Corollary 4.3.18 we even only have to check one representative of each equivalence class containing weight functions that do not allocate the same weight to several vertices of different pseudo classes.

Let $(v_1, \dots, v_n)$ be a fixed ordering of V in which vertices of the same pseudo class appear sequentially. We define Ω to be the set of all weight functions $\omega\colon V \to [k]$ for which the weights also appear sequentially in the weight vector $\big(\omega(v_1), \dots, \omega(v_n)\big) \in [k]^n$ and which do not allocate same weights to different pseudo classes. Now it is sufficient to check every function of Ω to decide whether G is k-equidominating. This holds since every k-equidominating function not in Ω has an equivalent weight function in Ω: we simply have to re-sort the weights within each pseudo class to obtain an equivalent weight function that lies in Ω.

In a second step, we check for each $\omega \in \Omega$ if it is a k-equidominating function. For that, let $\omega \in \Omega$, $D \subseteq V$ be an arbitrary minimal dominating set. We define $t_\omega := \omega(D)$ and $X_\omega := \{x \in \mathbb{Z}^k \mid 0 \le x_i \le |\omega^{-1}(i)|,\ i = 1, \dots, n\}$. Every vector $x \in X_\omega$ encodes a family of subsets of vertices $\mathcal{S}_\omega(x) \subseteq \mathcal{P}(V)$ (with $\mathcal{P}(V)$ being the power set of V) in the following sense: The value x_i of the i-th coordinate of x is equal to the vertices of weight i being in a subset, that is $\mathcal{S}_\omega(x) = \{S \subseteq V \mid |S \cap \omega^{-1}(i)| = x_i\}$. By Lemma 4.3.19, either every element of $\mathcal{S}_\omega(x)$ is a minimal dominating set or none. In fact, implicitly we have defined an equivalent relation on $\mathcal{P}(V)$ with $\mathcal{S}_\omega(x)$ being the equivalence classes ($x \in X_\omega$). Furthermore, by definition every element of $\mathcal{S}_\omega(x)$ has the same total weight $\sum_{i=1}^k i x_i = t_\omega$. The last thing to do is to check for each $x \in X_w$ whether x encodes minimal dominating sets if and only if $\sum_{i=1}^k i x_i = t_\omega$. If so, (ω, t_ω)

is a k-equidominating structure and hence G is k-equidominating. In conclusion, we achieve Algorithm 2.

Let us analyze the running time of the above-described algorithm. The determination of the pseudo classes partition in line 1 needs $\mathcal{O}(n+m+nm^2)$ time (compare Section 4.3). An arbitrary minimal dominating set D can be calculated straightforwardly in time $\mathcal{O}(n+m)$. We can describe a weight function by its weight vector regarding the fixed ordering. A basic, combinatorial result states that there are at most $\binom{n+k-1}{n}$ possibilities to divide the fixed ordered vertices $(v_1, \ldots, v_n)$ into k (possibly empty) intervals, where an interval only contains consequently in the ordering appearing vertices. There are $k!$ ways to distribute k weights one-to-one to the intervals, which leads to $|\Omega| \leq k!\binom{n+k-1}{n}$. With $\hat{n} := \max\{n, k^2\}$ we can further estimate the number of potential k-equidominating functions to be $|\Omega| = \mathcal{O}(\hat{n}^k/k)$ and Ω can be computed in time $\mathcal{O}((k\hat{n})^k)$ (for more details see [36]). The lines 8 and 10 can be executed in time $\mathcal{O}(n)$. The number of vectors in X_ω is $\Pi_{i=1}^k(|\omega^{-1}(i)|+1)$, which is bounded by $(n/k+1)^k = \mathcal{O}((\hat{n}/k)^k)$ (see Lemma 4 in [36]). The sums in line 12 and 15 are calculated in time $\mathcal{O}(k)$. Finally, it takes time $\mathcal{O}(n^2)$ to check if a subset of V is a minimal dominating set.

Taken together, we obtain a total running time of Algorithm 2 of

$$\begin{aligned}
&\mathcal{O}\left(n+m+nm^2+(\hat{n}k)^k+\frac{\hat{n}^k}{k}\left(n+\left(\frac{\hat{n}}{k}\right)^k\left(n+n^2+k\right)\right)\right)\\
&\quad=\mathcal{O}\left(nm^2+(\hat{n}k)^k+\frac{\hat{n}^{2k}n^2}{k^{k+1}}+\frac{\hat{n}^{2k}}{k^k}\right)\\
&\quad=\mathcal{O}\left(nm^2+n^kk^k+\frac{n^{2k+2}}{k^{k+1}}+\frac{n^{2k}k}{k^{k+1}}+k^{3k}+\frac{k^{4k}n^2}{k^{k+1}}+\frac{k^{4k}}{k^k}\right)\\
&\quad=\mathcal{O}\left(nm^2+n^kk^k+n^{2k+2}k^{-k-1}+k^{3k+3}\right).
\end{aligned}$$

□

The vertices of clique classes, clique bundles and stable set bundles must have the same weight with respect to any equidominating function (compare Section 4.3). Even though we know this in advance, it does not help us to speed up the (theoretical) running time of Algorithm 2, as there do not need to exist any twin class of cardinality at least two at all. This is why we do not make use of it in Theorem 4.4.1. However, one should definitely use this information for any implementation (as well as the stop criterion), as already mentioned at the end of the previous section.

The **domination number** of a graph is defined as the minimum cardinality of all dominating sets while the **upper domination number** equals the maximum cardinality of all dominating sets. Basically, the algorithm considers all subsets of vertices. Therefore it can be easily modified to provide the domination number and the upper domination number as well as minimal dominating sets of minimum and maximum cardinality.

If we modify Algorithm 2 slightly, then we can also use it to solve the TARGET-t EQUIDOMINATION problem.

Corollary 4.4.2. *For $t \in \mathbb{N}$, it is decidable whether a graph $G = (V, E)$ is target-t equidominating in time $\mathcal{O}\left(nm^2 + n^t t^t + n^{2t+2} t^{-t-1} + t^{3t+3}\right)$, with $|V| = n$ and $|E| = m$. Furthermore, a target-t equidominating structure is computed in this time if G is target-t equidominating.*

Proof. We simply set $k := t$ and instead of defining $t_\omega := \omega(D)$ in line 8 we proceed only with those weight functions $\omega \in \Omega$ for which $\omega(D) = t$ holds. □

Chapter 5

Fixed-parameter Tractability Results

In this chapter, we deduce complexity results for the two parameterized versions of the Equidomination problem. We show that the Target-t Equidomination and the k-Equidomination problem are fixed-parameter tractable in Section 5.2 and Section 5.3, respectively. For that, we use three reduction rules, which we examine in Section 5.1. In Section 5.4, we introduce a generalization of the k-Equidomination problem and prove that this problem also lies in FPT.

5.1 Reduction Rules

In this section, we examine three reduction rules. We use them to construct (generalized) kernels of the Target-t Equidomination problem as well as the k-Equidomination problem. The three reduction rules are related to each other. They are all based on the fact that there are vertices which – roughly speaking – play the same role regarding minimal dominating sets. For example, it does not matter which vertex of a clique class is in a minimal dominating set. We show that above a certain number it is not relevant how many of such vertices exist for being target-t equidominating or k-equidominating. Therefore, we can reduce the number of vertices to that certain number. However, to prove the third reduction rule, which concerns clique bundles, we have to use more elaborated tools and work with a sort of abstraction of graphs.

We call a graph **target-t k-equidominating** if there is a k-equidominating structure with target value t. Note that this is stronger than being both k-equidominating and target-t equidominating. For example, consider the star graph $K_{1,n}$ for some $n \in \mathbb{N}$. It is k-equidominating for every $k \geq n$ and target-t equidominating for every $t \geq n$. But it is not target-$(n'+1)$ n'-equidominating for any $n' \geq n$.

The first two reduction rules concern clique classes and stable set bundles. As clique classes and stable set bundles behave similarly, so do the reduction rules and their proofs to show that we can use them for a kernelization of the parameterized problems. Therefore, with the next lemma we first prove the general case.

Lemma 5.1.1. *Let G be a graph, r, $k \in \mathbb{N}$ and $M \subseteq V(G)$ a subset of pairwise mds-exchangeable vertices with $|M| > r$. Furthermore, let G' be the graph obtained from G by deleting all but r vertices of M. If*

(i) $|D \cap M| \leq r$ for every minimal dominating set $D \subseteq V(G)$ of G,

(ii) the vertices of $M \cap V(G')$ are pairwise mds-exchangeable, and

(iii) every dominating set $D \subseteq V(G')$ of G' is a dominating set of G,

then the following equivalence holds for all $t \leq r$:

$$G \text{ is target-}t\ k\text{-equidominating} \Longleftrightarrow G' \text{ is target-}t\ k\text{-equidominating}.$$

Proof. To start, we claim that a subset $D \subseteq V(G')$ is a minimal dominating set of G' if and only if D is a minimal dominating set of G. For that, let $D \subseteq V(G')$.

First, let D be a minimal dominating set of G'. By assumption (iii), D is a dominating set of G. Suppose that D is not minimal in G. Then there is a proper subset $\widetilde{D} \subsetneq D$ that is a dominating set of G. Since G' is an induced subgraph of G, $\widetilde{D}$ is a dominating set of G', a contradiction.

Secondly, let D be a minimal dominating set of G. Again it follows that D is a dominating set of G'. Suppose D is not minimal in G'. Then there is a proper subset $\widetilde{D} \subsetneq D$ that is a minimal dominating set of G'. It follows from the preceding paragraph that $\widetilde{D}$ is also a minimal dominating set of G. This contradicts the minimality of D in G and the claim is proved.

Now, we prove the equivalence stated in the lemma. Let $t \leq r$ and $M' := M \cap V(G')$.

$\underline{\Longrightarrow}$: Let G be target-t k-equidominating and (ω, t) be a k-equidominating structure of G. Any subset of $V(G')$ is a minimal dominating set of G' if and only if it is a minimal dominating set of G, which in turn is the case if and only if it has total weight t. It follows that (ω', t) with $\omega' := \omega\big|_{V(G')}$ is a k-equidominating structure of G' with target value t.

$\underline{\Longleftarrow}$: Let G' be target-t k-equidominating and (ω', t) be a k-equidominating structure of G'. We define

$$\omega(v) := \begin{cases} \omega'(v), & \text{if } v \in V(G'), \\ \omega'(w), & \text{otherwise,} \end{cases}$$

for any $w \in M \cap V(G')$. Since every equidominating function is constant on a set of pairwise mds-exchangeable vertices, the choice of w is irrelevant for the definition of ω. We show that (ω, t) is a k-equidominating structure of G. It is clear that $\omega(v) \leq k$ for every $v \in V(G)$.

Let $X \subseteq V(G)$ be a subset of vertices of G with $\omega(X) = t$. As $\omega(X \cap M) \leq \omega(X) = t$, we get $|X \cap M| \leq t$. Therefore, we can exchange vertices of $X \setminus V(G')$, if any, with vertices of $M' \setminus X$ (one to one) to obtain a subset $\widetilde{X} \subseteq V(G')$ with $\omega'(\widetilde{X}) = \omega(\widetilde{X}) = t$.

It follows that $\widetilde{X}$ is a minimal dominating set of G' and thus of G, too. To construct $\widetilde{X}$, we only exchanged vertices with each other that are mds-exchangeable. It follows that X is a minimal dominating set of G.

Now, let $D \subseteq V(G)$ be a minimal dominating set of G. By assumption (i), we have $|D \cap M| \leq r$. As before, we can exchange vertices of $D \setminus V(G')$ with vertices of $M' \setminus D$ to get a subset $\widetilde{D} \subseteq V(G')$ that is a minimal dominating set of G and thus of G'. This means $\omega(\widetilde{D}) = t$ and since ω is constant on M, we get $\omega(D) = t$. □

The formulation of Lemma 5.1.1 seems a bit complicated: why are we working with some $r \in \mathbb{N}$ and $t \leq r$ instead of taking a (fixed) $t \in \mathbb{N}$ from the beginning? Indeed, for the Target-t Equidominating problem we could do so. However, in the proof of Theorem 5.3.1, in which we show that the k-Equidominating problem is fixed-parameter tractable, we can only determine an upper bound for the potential target value. Therefore, we need Lemma 5.1.1 with a range of possible target values.

The first reduction rule is about reducing the vertices of a clique class to a certain number $r \in \mathbb{N}$.

> ***r*-Clique Class Reduction**: If a clique class C contains more than r vertices, delete all but r vertices of C.

The next lemma shows that this rule can be used to construct kernels for the parameterized problems.

Lemma 5.1.2. *Let G be a graph, r, $k \in \mathbb{N}$ and $C \subseteq V(G)$ a clique class with $|C| > r$. Furthermore, let G' be the graph obtained from G by applying the r-Clique Class Reduction rule with respect to C. Then for all $t \leq r$, the graph G is target-t k-equidominating if and only if G' is target-t k-equidominating.*

Proof. The vertices of a clique class are pairwise mds-exchangeable, both in G and G'. Further, we know that $|D \cap C| \leq 1$ for every minimal dominating set $D \subseteq V(G)$ of G, by Observation 4.3.4. We define $C' := C \cap V(G')$. Let $D \subseteq V(G')$ be a dominating set of G'. The vertex (or vertices) of D that dominates C' also dominates the vertices of $C \setminus C'$ in G. So, D is a dominating set of G and we can apply Lemma 5.1.1. This finishes the proof. □

One might ask why we cannot reduce the number of vertices of a clique class simply to one. Analogously to the notion of Lemma 5.1.2, let G' be the graph obtained from G by deleting all but one vertex $v \in C$ of the clique class. Indeed, G' is equidominating if G is. Also, if G' is equidominating, then we can show that every minimal dominating set of G has the same total weight. However, we cannot ensure in this situation that every subset of that total weight is a minimal dominating set. This is because the weight of v exists several times in G but only once in G'. As a consequence, at least t vertices (having a total weight of at least t) of a large clique class must remain to obtain the equivalence of being target-t k-equidominating. The same applies for the upcoming two reduction rules for stable set bundles and clique bundles. Actually, we have already seen in Remark 3.2.5 that we cannot reduce a clique class to one vertex since adding a true twin does not preserve equidomination in general.

The next rule is about stable set bundles. As seen before, there can be arbitrarily large stable set bundles in an equidominating graph. In fact, every stable set bundle itself is an equidominating graph (see Lemma 3.3.1). Again, a positive integer $r \in \mathbb{N}$ specifies the reduction rule.

> **r-Stable Set Bundle Reduction**: If a stable set bundle $\mathcal{S}$ contains more than r stable set classes, delete all but r stable set classes of $\mathcal{S}$.

The upcoming lemma shows that the r-Stable Set Bundle Reduction rule can be used to obtain kernels for the parameterized problems.

Lemma 5.1.3. *Let G be a graph, $r, k \in \mathbb{N}$ and $\mathcal{S} \subseteq V(G)$ a stable set bundle containing more than r stable set classes. Further, let G' be the graph obtained from G by applying the r-Stable Set Bundle Reduction rule with respect to $\mathcal{S}$. Then for all $t \leq 2r$, the graph G is target-t k-equidominating if and only if G' is target-t k-equidominating.*

Proof. Note that there are $2r$ vertices in $\mathcal{S} \cap V(G')$. First, we consider the case $r = 1$. In this case, $\mathcal{S}$ becomes a stable set class in G'. Nevertheless, the lemma holds. There are only the two possible values for t: $t = 1$ or $t = 2$. Neither G nor G' are target-1 k-equidominating as the complete graphs K_n ($n \in \mathbb{N}$) are the only target-1 k-equidominating graphs (with equidominating structure $(\omega \equiv 1, 1)$).

For $t = 2$, there can exist only one more pseudo class besides $\mathcal{S}$, otherwise G and G' are not target-2 k-equidominating, by Corollary 4.3.18. If this second pseudo class is an adjacent singleton class or an adjacent clique class, then both G and G' are target-2 k-equidominating. In the other cases neither G nor G' are target-2 k-equidominating since then a minimal dominating set exists with more than two vertices.

Secondly, let $r \geq 2$. Again, we show that all conditions of Lemma 5.1.1 are met. Following Lemma 4.3.13, the vertices of $\mathcal{S}$ are pairwise mds-exchangeable. The same holds for the vertices of $\mathcal{S} \cap V(G')$ since at least two stable set classes of $\mathcal{S}$ remain in G'. It is easy to see that $|D \cap \mathcal{S}| \leq 2$ for every minimal dominating set $D \subseteq V(G)$ of G. Finally, we can show, analogously to the proof of Lemma 5.1.2, that every minimal dominating set of G' is a dominating set of G. □

Note that the condition of Lemma 5.1.1(ii) is only fulfilled if we delete the stable set classes of $\mathcal{S}$ completely, and not only one of the two vertices of a stable set class. However, we ensure this by the formulation of the r-Stable Set Bundle Reduction rule. Thus, Lemma 5.1.1 can be applied.

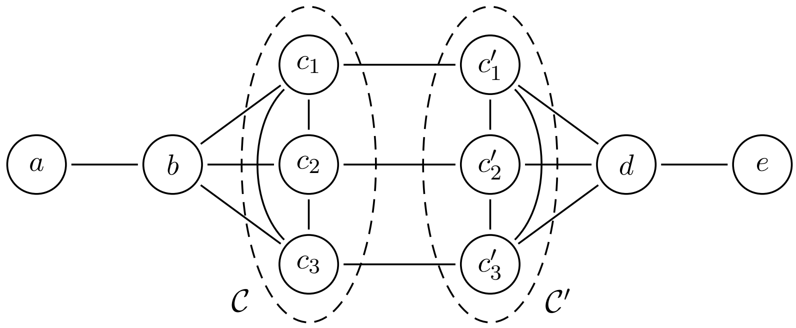


Figure 5.1: A (non-equidominating) graph on 10 vertices; $\mathcal{C}$ and $\mathcal{C}'$ are clique bundles consisting of three singleton classes and the minimal dominating set $\{a, c_1, c_2, c_3, e\}$ contains every vertex of $\mathcal{C}$.

The last reduction rules considers clique bundles. As we already know, the vertices of a clique bundle can have different neighborhoods. Thus, even though we can bound the number of pseudo classes of a graph, there can be a large number of twin classes (more precisely, singleton classes and clique classes) in a clique bundle. In the case of more than one clique bundle being in a graph, there can be arbitrarily many distinct neighborhoods, and consequently, arbitrarily many twin classes in a clique bundle. For example, the graph in Figure 5.1 can be extended to any number of singleton classes in both clique bundles: we just add adjacent vertices c_i and c_i' ($i \geq 4$) to the clique bundles $\mathcal{C}$ and $\mathcal{C}'$, respectively, analogously to the existing vertices. The same holds for the graph shown in Figure 4.3 on page 47. Therefore, it is not enough to bound the clique classes of a clique bundle using the Clique Class Reduction rule.

Furthermore, a special case can occur: if a clique bundle only contains singleton classes, then more than one vertex of such a clique bundle can be in a minimal dominating set. Figure 5.1 shows a graph, where even every vertex of a clique bundle is contained in the same minimal dominating set. However, if there is at least one clique class in a clique bundle, then at most one vertex of the clique bundle is in a minimal dominating set. Otherwise, we could exchange vertices such that two vertices of the same clique class are in a minimal dominating set, a contradiction.

These two facts make it harder to bound the number of vertices in a clique bundle in terms of the parameters k and t as in the case of clique classes or stable set bundles. The keynote to overcome this is gathering the vertices of a clique bundle into certain subsets. In such a subset, the vertices have the same neighborhood regarding all pseudo classes except clique bundles. With respect to clique bundles, however, the vertices of a subset have the same number of neighbors in each clique bundle.

To formalize the above-mentioned idea, we introduce the following notion.

Definition 5.1.4. *Let G be a graph with pseudo class partition $\{P_1, \ldots, P_s\}$. For every vertex $v \in V(G)$ we define the vector $\mu^v = (\mu_1^v, \ldots, \mu_s^v) \in \mathbb{N}_0^s$ as follows:*

if $v \in P_i$, then we set

$$\mu_i^v := \begin{cases} 1, & \text{if } P_i \text{ is a singleton class, clique class or clique bundle,} \quad (5.1a)\\ 2, & \text{if } P_i \text{ is a stable set bundle,} \quad (5.1b)\\ |P_i|, & \text{if } P_i \text{ is a stable set class.} \quad (5.1c) \end{cases}$$

If $v \notin P_i$ and P_i is not a clique bundle, then we set

$$\mu_i^v := \begin{cases} 1, & \text{if } v \text{ is adjacent to } P_i, \quad (5.2a)\\ 0, & \text{if } v \text{ is not adjacent to } P_i. \quad (5.2b) \end{cases}$$

If $v \notin P_i$ and P_i is a clique bundle, then we set

$$\mu_i^v := \begin{cases} |P_i \setminus N[v]| + 1, & \text{if there exists a minimal dominating set } D \subseteq V \text{ with } N[v] \cap D \subseteq P_i, \quad (5.3a)\\ 0, & \text{otherwise.} \quad (5.3b) \end{cases}$$

*We call μ^v the **mds-vector** of v.*

We remark that μ^v rather contains information about how v can be dominated by the pseudo classes of a graph, than how v dominates the pseudo classes (in particular in the cases (5.3a) and (5.3b)).

Let P_i be a clique bundle and D a minimal dominating set such that $N[v] \cap D \subseteq P_i$. Then, v is dominated only by vertices of P_i. Furthermore, the vertices of P_i are pairwise mds-exchangeable. This means that there are more vertices in $D \cap P_i$ than in $P_i \setminus N[v]$. So, the number μ_i^v tells us how many vertices of P_i must be at least in a minimal dominating set to dominate v, such that v is dominated only by vertices of P_i.

To decide whether a minimal dominating set D with $N[v] \cap D \subseteq P_i$ exists, it is sufficient to check if $(V(G) \setminus N[v]) \cup P_i$ is a dominating set. If so, this dominating set contains a minimal dominating set $D \subseteq (V(G) \setminus N[v]) \cup P_i$ with $N[v] \cap D \subseteq P_i$.

The values of μ^v are bounded by t in every target-t equidominating graph.

Lemma 5.1.5. *Let $G = (V, E)$ be a graph with pseudo class partition $\{P_1, \dots, P_s\}$ and let $r \in \mathbb{N}$. If there is a vertex $v \in V$ with $\mu_i^v > r$ for some $i \in [s]$, then G is not target-t equidominating for all $t \leq r$.*

Proof. Let $t \leq r$ and $\mu_i^v > r$. If $v \in P_i$ and P_i is a stable set bundle, then $\mu_i^v = 2$ and hence $r = 1$. Since we can extend two non-adjacent vertices of P_i to a minimal dominating set (with at least two elements), G is not target-1 equidominating. Analogously, if $v \in P_i$ and P_i is a stable set class, there exists a minimal dominating set containing more than t vertices.

The last possible case for $\mu_i^v > 1$ is $v \notin P_i$ and P_i is a clique bundle. Then, there exists a minimal dominating set D with $N[v] \cap D \subseteq P_i$ and $|D| \geq |D \cap P_i| \geq \mu_i^v > t$. Again, G is not target-t equidominating. □

As mentioned before, we want to gather – or rather partition – the vertices of clique bundles into subsets. We do this in a way such that the vertices of each subset have identical mds-vectors. However, a problem arises if we then reduce such a subset of a graph. Likewise for the other reduction rules, the main condition to prove that the reduction is safe for the parameterized problems is: a subset of vertices of the reduced graph is a minimal dominating set if and only if it is a minimal dominating set of the original graph (compare Lemma 5.1.1(iii)).

Now the problem is the following: it can occur that the pseudo class partition changes if we delete some vertices of a clique bundle. For example, if we delete c_3 of the graph shown in Figure 5.1, then c_3' is not part of the clique bundle $\mathcal{C}'$ anymore and becomes a singleton class instead. Clearly, if the pseudo class partition changes, then we also obtain different mds-vectors. With possibly different pseudo class partitions and mds-vectors we cannot prove the above-mentioned main condition.

Since the clique bundles are determined by the graph, we need a more general structure. Therefore, we introduce a new mathematical object and transfer all relevant information and properties to it. However, we do this in a way such that neither the pseudo class partition nor the mds-vectors change when we delete some elements from it. Roughly speaking, the graph does not determine the partition, but the partition determines the graph.

Definition 5.1.6. *A **pseudo graph** is a triple $\mathfrak{P} = (V, \mathcal{P}, \mu)$, consisting of a non-empty set V, a partition $\mathcal{P} = \{P_1, \dots, P_s\}$ of V and a function $\mu\colon V \to \mathbb{N}_0^s$, such that $\mu(v) \neq (0, \dots, 0)$ for all $v \in V$.*

Next, we need an equivalent for (minimal) dominating sets in pseudo graphs. By $(\mu(v))_i$ we denote the i-th component of $\mu(v)$.

Definition 5.1.7. *Let $\mathfrak{P} = (V, \mathcal{P} = \{P_1, \dots, P_s\}, \mu)$ be a pseudo graph and $X \subseteq V$. We call X a **dense set of $\mathfrak{P}$** if for every $v \in V$ a block $P_i \in \mathcal{P}$ exists with $0 < (\mu(v))_i \leq |X \cap P_i|$. If every proper subset $X' \subsetneq X$ is not dense, then X is said to be a **minimal dense set of $\mathfrak{P}$**.*

If it is not required for each $v \in V$ that at least one component of $\mu(v)$ is greater than zero (see Definition 5.1.6), then it is possible that no (minimal) dense set exists at all. The next definition motivates the two previous definitions.

Definition 5.1.8. *Let $G = (V, E)$ be a graph with pseudo class partition $\{P_1, \dots, P_s\}$ and mds-vectors $\mu^v \in \mathbb{N}_0^s$ for $v \in V$. By $\mathfrak{P}(G) = (V, \{P_1, \dots, P_s\}, \mu)$ we denote the **pseudo graph of G**, where $\mu(v) := \mu^v$ for each $v \in V$. We say that a pseudo graph $\mathfrak{P}$ is **induced by** a graph G if $\mathfrak{P}(G) = \mathfrak{P}$.*

Note that the pseudo graph of a graph is unique (up to the order of the pseudo class partition), while two different graphs can have the same pseudo graph. For example, two graphs have identical pseudo graphs if they differ only with respect to edges between a clique bundle $\mathcal{C}$ and a vertex v where there is no minimal dominating set D with $N[v] \cap D \subseteq \mathcal{C}$. Further, there are pseudo graphs that are not induced by a graph.

The pseudo graph of a graph with n vertices and m edges can be calculated in time $\mathcal{O}(nm^2 + n^2)$. By Corollary 4.3.21, we can compute the pseudo class partition in time $\mathcal{O}(nm^2)$. Furthermore, we can determine all mds-vectors in time $\mathcal{O}(n^2 + nm)$.

The next lemma and the subsequent corollary show that induced pseudo graphs and minimal dense sets indeed correspond to minimal dominating sets as desired.

Lemma 5.1.9. *Let $G = (V, E)$ be a graph with mds-vectors μ^v, $v \in V$, pseudo class partition $\mathcal{P} = \{P_1, \ldots, P_s\}$ and pseudo graph $\mathfrak{P}(G)$. Further, let $D \subseteq V$. Then D is a dominating set of G if and only if D is a dense set of $\mathfrak{P}(G)$.*

Proof. $\Longrightarrow$: Let $D \subseteq V$ be a dominating set of G and $v \in V$ with $v \in P_l$ ($l \in [s]$). If P_l is a stable set class, then every vertex of P_l or a neighbor $x \in P_i \cap N(v)$ lies in D. In the first case we have $0 < \mu^v_l = |P_l| = |D \cap P_l|$ (see (5.1c) in Definition 5.1.4). In the latter case, if there is a vertex $w \in N[v] \cap D$ that lies in a singleton class, a clique class, a stable set class or a stable set bundle P_i ($i \in [s]$), then we have $0 < \mu^v_i = 1 \leq |D \cap P_i|$, by (5.2b). If each vertex of $N[v] \cap D$ lies in a clique bundle, then we cannot exchange (within each clique bundle) all neighbors of v lying in D with non-neighbor of v not lying in D, (this would contradict the mds-exchangeability). This means that there is a clique bundle P_i with $0 < \mu^v_i \leq |D \cap P_i|$ (compare (5.3a)).

If P_l is a singleton class, a stable set class, a clique class or a clique bundle, we can show analogously that $0 < \mu^v_i = |D \cap P_i|$ holds for some $i \in [s]$, by considering (5.1a) and (5.1b). Hence, D is a dense set of $\mathfrak{P}(G)$.

$\Longleftarrow$: Let D be a dense set of $\mathfrak{P}$ and $v \in V$ with $v \in P_l$ ($l \in [s]$). Then, there exists $P_i \in \mathcal{P}$ with $0 < (\mu(v))_i \leq |D \cap P_i|$. If $i = l$, then v or a vertex of $N(v) \cap P_l$ lies in D, by (5.1). Thus, v is dominated. If $i \neq l$ and P_i is a singleton class, stable set class, clique class or a stable set bundle (in G), then v is dominated by the vertices of $D \cap P_i$, by (5.2b). If P_i is a clique bundle, then v has less non-neighbors in P_i than there are vertices in $D \cap P_i$ (since $|P_i \setminus N[v]| + 1 \leq |D \cap P_i|$, by (5.3a)). This means, at least one vertex of $N(v) \cap P_i$ lies in D. Hence, v is dominated. It follows that D is a dominating set of G and the proof is finished. □

Using Lemma 5.1.9, it is straightforward to prove the following corollary by contradiction.

Corollary 5.1.10. *Let $G = (V, E)$ be a graph with pseudo graph $\mathfrak{P}(G)$ and let $D \subseteq V$. Then D is a minimal dominating set of G if and only if D is a minimal dense set of $\mathfrak{P}(G)$.*

Next, we introduce analogous terms to identify the minimal dense sets of a pseudo graph.

Definition 5.1.11. *A pseudo graph $\mathfrak{P} = (V, \mathcal{P}, \mu)$ is called **equidense** if there exists $t \in \mathbb{N}$ and a weight function $\omega\colon V \to \mathbb{N}$ such that for all $D \subseteq V$ the following equivalence holds:*

$$D \text{ is a minimal dense set} \iff \omega(D) = t\,.$$

*Further, we call the pair (ω, t) an **equidense structure**, ω an **equidense function** and t a **target value**.*

Definition 5.1.12. *For a given $t \in \mathbb{N}$, a pseudo graph $\mathfrak{P} = (V, \mathcal{P}, \mu)$ is called **target-t equidense** if there exists an equidense structure of the form (ω, t) of G.*

Definition 5.1.13. *For a given $k \in \mathbb{N}$, a pseudo graph $\mathfrak{P} = (V, \mathcal{P}, \mu)$ is said to be **k-equidense** if there exists an equidense structure (ω, t) with $\omega\colon V \to [k]$ for some $t \in \mathbb{N}$. In this case, (ω, t) is called a **k-equidense structure** and ω a **k-equidense function**.*

Finally, we call a pseudo graph **target-*t* *k*-equidense** if a k-equidense structure with target value t exists. By Corollary 5.1.10, we immediately get the following result.

Corollary 5.1.14. *Let G be a graph with pseudo graph $\mathfrak{P}(G)$ and let k, $t \in \mathbb{N}$. Then G is target-t k-equidominating if and only if $\mathfrak{P}(G)$ is target-t k-equidense. Moreover, in the affirmative case, we can use the same structure to identify minimal dominating and minimal dense sets in G and $\mathfrak{P}(G)$, respectively.*

Now, we gathered together everything to define the reduction rule and to prove that we can use it (in combination with Corollary 5.1.14) for the TARGET-t EQUIDOMINATION problem and the k-EQUIDOMINATION problem. Again, the rule is specified by a positive integer $r \in \mathbb{N}$.

> ***r*-Pseudo Graph Reduction**: If a subset $M \subseteq P$ of a block P of the partition of a pseudo graph with $\mu(v) = \mu(w)$ for all v, $w \in M$ contains more than r vertices, delete all but r vertices of M.

Lemma 5.1.15. *Let r, $k \in \mathbb{N}$ and $\mathfrak{P} = (V, \mathcal{P} = \{P_1, \dots, P_s\}, \mu)$ be a pseudo graph induced by a graph $G = (V, E)$, with $(\mu(v))_i \leq r$ for all $v \in V$ and $i \in [s]$. Let P_l ($l \in [s]$) be a block of $\mathcal{P}$ such that P_l is a clique bundle of G. Further, let $M \subseteq P_l$ be a subset of P_l with $\mu(v) = \mu(w)$ for all v, $w \in M$ and $|M| > r$.*

Let $\mathfrak{P}' = (V', \mathcal{P}', \mu')$ be the pseudo graph obtained from $\mathfrak{P}$ by applying the r-Pseudo Graph Reduction rule with respect to M. Then for all $t \leq r$, $\mathfrak{P}$ is target-t k-equidense if and only if $\mathfrak{P}'$ is target-t k-equidense.

Proof. Note that $\mu' = \mu\big|_{V'}$ and that besides P_l the partitions $\mathcal{P}$ and $\mathcal{P}'$ of V and V', respectively, have identical blocks.

In the first place, we claim that a subset $D \subseteq V'$ is a minimal dense set of $\mathfrak{P}'$ if and only if D is a minimal dense set of $\mathfrak{P}$. To prove this, it is sufficient to show that the equivalence holds for dense sets (not necessarily minimal). So, let $D \subseteq V'$. First, let D be a dense set of $\mathfrak{P}'$. Since $\mu(v) = \mu(w)$ for all $v \in P_l \setminus P_l'$ and $w \in P_l'$, we directly get that D is a dense set of $\mathfrak{P}$. Secondly, let D be a dense set of $\mathfrak{P}$. By considering the definition of dense, D is clearly a dense set of $\mathfrak{P}'$ and the claim is proved.

Now, we prove the equivalence stated in the lemma. Let $t \leq r$ and $M' = M \cap V'$.

$\Longrightarrow$: Let $\mathfrak{P}$ be target-t k-equidense and (ω, t) be a k-equidense structure of $\mathfrak{P}$. Any subset of V' is a minimal dense set of $\mathfrak{P}'$ if and only if it is a minimal dense set of $\mathfrak{P}$, which in turn is the case if and only if it has total weight t. It follows that (ω', t) with $\omega' := \omega\big|_{V'}$ is a k-equidense structure of $\mathfrak{P}'$ with target value t.

$\Longleftarrow$: Let $\mathfrak{P}'$ be target-t k-equidense and (ω', t) be a k-equidense structure of $\mathfrak{P}'$. Since $\mathfrak{P}$ is induced by a graph and P_l is a clique bundle of G, there exists a minimal dense set D with $|D \cap P_l| = 1$. It follows that ω' is constant on $P_l \cap V'$. We define

$$\omega(v) := \begin{cases} \omega'(v), & \text{if } v \in V', \\ \omega'(w), & \text{otherwise,} \end{cases}$$

with any $w \in P_l \cap V'$ and claim that (ω, t) is a k-equidense structure of $\mathcal{P}$. It is clear that $\omega(v) \leq k$ for every $v \in V$.

First, let $X \subseteq V$ be a subset with $\omega(X) = t$. As $\omega(X \cap M) \leq \omega(X) = t$, we get $|X \cap M| \leq t \leq r$. Since $|M'| = r$, we can assume $|X \cap M| \subseteq M'$ and hence $X \subseteq V'$. It follows that X is a minimal dense set of $\mathfrak{P}'$ and thus also of $\mathfrak{P}$.

Secondly, let $D \subseteq V(G)$ be a minimal dense set of $\mathfrak{P}$. Since $(\mu(v))_i \leq r$ for all $v \in V$ and $i \in [s]$, we have $|D \cap M| \leq |D \cap P_l| \leq r$. Again, we can assume that $D \subseteq V'$. This means that D is a minimal dense set of $\mathfrak{P}'$. It follows that $\omega'(D) = t$ and consequently $\omega(D) = t$. This finishes the proof. □

We remark that the vertices of a clique class (that is contained in a clique bundle) have identical mds-vectors. This means that every block of $\mathfrak{P}$ that is a clique class of G is entirely contained in one of such subsets M. Using Lemma 5.1.5, we can bound the number of distinct mds-vectors in a clique bundle of a graph for which we have a bound of possible target values. Together with Lemma 5.1.15, we can bound the number of elements of a block induced by a clique bundle.

In Appendix A we state two reduction rules that can be applied to clique bundles $\mathcal{C}$ for which $|D \cap \mathcal{C}| \leq 1$ holds for every minimal dominating set D. This is the case, for example, if a clique bundle contains at least one clique class (and not only singleton classes). Even though these reductions are not useful for our (theoretical) results, they might speed up implementations of the discussed algorithms.

5.2 Target-t Equidomination

We will now prove that the Target-t Equidomination problem is fixed-parameter tractable. For that, we construct a generalized kernel the size of which can be bounded by a function of t. We apply the *XP* algorithm described in Section 4.4 to the kernel. In this way, we do not only obtain a complexity result but also an explicit *FPT* algorithm.

We begin with the Target-t Equidomination problem since its *FPT* result is easier to obtain. The main reason for that lies in the fact that we can bound the size of a stable set class in a target-t equidominating graph by t (compare Observation 4.3.5). This is not the case in a k-equidominating graph. For example, the edgeless graph $\overline{K_n}$ is even 1-equidominating for all $n \in \mathbb{N}$.

Theorem 5.2.1. *The* Target-t Equidomination *problem admits a generalized kernel of size $\mathcal{O}\left(t^{t+1}\right)$ which is computable in polynomial time. Furthermore, there exists an $\mathcal{O}\left(nm^2 + n^2 + t^{2t^2+3t+1}\right)$ time algorithm to solve the* Target-t Equidomination *problem for a graph on n vertices and m edges.*

Proof. Let G be a graph with $|V(G)| = n$, $|E(G)| = m$ and $t \in \mathbb{N}$. First, we use an algorithm for modular decomposition to compute the twin partition. If there is a stable set class with more than t vertices, then G is not target-t equidominating, by Observation 4.3.5. If no such stable set class exists, then we apply Algorithm 1 to obtain the pseudo class partition $\{P_1, \ldots, P_s\}$ and we determine the mds-vectors μ^v for all $v \in V(G)$.

If $s > t$ or $\mu_i^v > t$ for some $v \in V(G)$ and $i \in [s]$, then we conclude that G is not target-t equidominating, by Corollary 4.3.18 and Lemma 5.1.5. Otherwise, we apply the r-Clique Class Reduction rule with $r = t$ and the r-Stable Set Bundle Reduction rule with $r = \lceil t/2 \rceil$ to all clique classes and stable set bundles, respectively, to obtain an induced subgraph G' of G. Then, we determine the pseudo graph $\mathfrak{P}(G')$ and apply the r-Pseudo Graph Reduction rule (simultaneously) to all blocks of $\mathfrak{P}(G')$ that are clique bundles of G', again with $r = t$. This yields a pseudo graph $\mathfrak{P}'' = (V'', \mathcal{P}'', \mu'')$.

Now, $\mathcal{P}''$ has at most t blocks. Every block that is a clique class or stable set class in G contains at most t elements. Blocks of stable set bundles may have up to $t+1$ elements. In blocks of clique bundles there can be at most $(t+1)^{(t-1)}$ distinct mds-vectors (note that $\mu^v(i) = 1$ for all vertices v within a clique bundle P_i). Thus, after the reductions every block of $\mathcal{P}''$ contains at most $t(t+1)^{(t-1)}$ elements.

Taken together, we get $n'' := |V''| = \mathcal{O}\left(t^{t+1}\right)$. Finally, by Lemma 5.1.2, Lemma 5.1.3 and Lemma 5.1.15, the obtained pseudo graph $\mathfrak{P}''$ is target-t equidense if and only if G is target-t equidominating.

We can compute the pseudo class partition of G in time $\mathcal{O}(nm^2)$. The computation of the mds-vectors μ^v for all $v \in V(G)$ and the partitioning of the clique bundles in preparation for the Pseudo Class Reduction can be done in $\mathcal{O}(n^2 + nm)$. Applying the three reduction rules needs linear time. This finishes the proof of the first statement of this theorem.

By the proof of Lemma 5.1.1, we know that we can easily extend any target-t equidominating structure of G' to a target-t equidominating structure of G. The same holds for a target-t equidense structure of $\mathfrak{P}'$ and $\mathfrak{P}$. Moreover, we can use the same function and target value for G' and $\mathfrak{P}'(G')$, by Corollary 5.1.14. It follows that an equidense structure of $\mathfrak{P}'$ induces an equidominating structure of G.

Furthermore, by the proofs of the lemmas of Section 5.1, we know that a subset $D \subseteq V''$ is a minimal dense set of $\mathfrak{P}''$ if and only if D is a minimal dominating set of G. Due to this equivalence, we can analogously apply Algorithm 2 to obtain a target-t equidense structure of $\mathfrak{P}''$, if existent.

Since we do not have to compute the pseudo class partition, the first two summands of the running time Algorithm 2 vanish. So, for applying Algorithm 2 to $\mathfrak{P}''$ we need time

$$\begin{aligned}\mathcal{O}\left(n''^t t^t + n''^{2t+2} t^{-t-1} + t^{3t+3}\right) &= \mathcal{O}\left(t^{t^2+2t} + t^{2t^2+4t+2} t^{-t-1} + t^{3t+3}\right)\\ &= \mathcal{O}\left(t^{2t^2+3t+1}\right) .\end{aligned}$$

Together with the computation of the pseudo class partition and the mds-vectors of G, we achieve a total running time of

$$\mathcal{O}\left(nm^2 + n^2 + nm + t^{2t^2+3t+1}\right) = \mathcal{O}\left(nm^2 + n^2 + t^{2t^2+3t+1}\right) .$$

□

We point out that – in contrast to an induced pseudo graph – Algorithm 2 cannot be used to find an equidense structure of an arbitrary pseudo graph. If a pseudo graph $\mathfrak{P} = (V, \mathcal{P}, \mu)$ is not induced by a graph, then it is possible, for example, that an element of $v \in V$ is not contained in any minimal dense set. Therefore, its weight could be greater than t. Furthermore, elements of different blocks of $\mathcal{P}$ can have equal weights. This means the central requirement of Algorithm 2 is not necessarily fulfilled.

5.3 k-Equidomination

In this section, we show that the k-Equidomination problem admits a generalized kernel the size of which is bounded by a function of k and, therefore, is fixed-parameter tractable. The proof of Theorem 5.3.1 is along the lines of the argumentation used in [36] to prove that the k-Equistable problem is fixed-parameter tractable.

In the following, we assume that $k > 1$. In a 1-equidominating graph there can only be one pseudo class. Thus, the only graphs that are 1-equidominating are the graphs

K_n, $\overline{K_n}$ and $T(2n, n)$ ($n \in \mathbb{N}$) with equidominating structure ($\omega \equiv 1, 1$), ($\omega \equiv 1, n$) and ($\omega \equiv 1, 2$), respectively.

Theorem 5.3.1. *The* k-EQUIDOMINATION *problem admits a generalized kernel of size* $\mathcal{O}(k^{3k+1})$ *which is computable in polynomial time. Furthermore, there is an algorithm to solve the* k-EQUIDOMINATION *problem which runs for a given graph on* n *vertices and* m *edges in time* $\mathcal{O}\left(nm^2 + n^2 + k^{6k^2+7k+1}\right)$.

The proof of this theorem builds upon two more lemmas. The first one deals with the problem that stable set classes are not of bounded size in a k-equidominating graph, which is the case for target-t equidominating graphs. We show, that if a large stable set class exists, then every other pseudo class is relatively small.

Lemma 5.3.2. *A graph* $G = (V, E)$ *is not* k*-equidominating* ($k \in \mathbb{N}$) *if* G *has two pseudo classes of size at least* k^2*, where one of those pseudo classes is a stable set class.*

Proof. For $k \in \mathbb{N}$, let $G = (V, E)$ be a graph with a stable set class $S \subseteq V$ and a different pseudo class $P \subseteq V$ such that $\min\{|S|, |P|\} \geq k^2$. Suppose that G is k-equidominating and let $\omega \colon V \to [k]$ be a k-equidominating function. It is straightforward that – regardless of what kind of pseudo class P is – there is a minimal dominating set D with $S \subseteq D$ and $|P \cap D| \leq 2$.

Let i, $j \in [k]$ be weights such that $|\{s \in S \mid \omega(s) = i\}| \geq k$ and $|\{p \in P \mid \omega(p) = j\}| \geq k$. Such weights exist due to the size of S and P. Further, at least k vertices of P of weight j are not in D. Now, let $S' \subseteq S$ be a subset of j vertices of weight i and $P' \subseteq P$ be a subset of i vertices of weight j with $P' \cap D = \emptyset$. Consequently, we get $\omega(S') = \omega(P')$. This leads to a contradiction as the set $(D \setminus S') \cup P'$ is not a minimal dominating set (see Observation 4.3.3) while being of the same total weight as the minimal dominating set D. □

The second lemma concerns relatively large, isolated stable set classes.

Lemma 5.3.3. *Let* $G = (V, E)$ *be a graph and* $k \in \mathbb{N}$*. Furthermore, let* S *be an isolated stable set class of size at least* k^5 *and* $|S \setminus V| \leq k^3$*. Then* G *is* k*-equidominating if and only if there exists a* k*-equidominating function that is constant on* S*.*

Proof. First note that the stable set class S is contained in every minimal dominating set of G. The sufficiency-part is trivial.

So, let G be k-equidominating with k-equidominating structure (ω, t). As there are only k different weights and $|S| \geq k^5$, a weight $i \in [k]$ exists such that $|S_i| \geq k^4$, with $S_i := \{s \in S \mid \omega(s) = i\}$. We show that (ω', t') with

$$\omega'(v) := \begin{cases} \omega(v), & \text{if } v \in V \setminus S, \\ i, & \text{if } v \in S, \end{cases}$$

and $t' := t - \omega(S) + \omega'(S) = t - \omega(S) + i|S|$ is a k-equidominating structure. It is easy to see that ω' is bounded by k and that $\omega'(D) = t'$ holds for every minimal dominating set D.

As before, the only tricky part is to show that any subset $X \subseteq V$ with $\omega'(X) = t'$ is a minimal dominating set. So, let $X \subseteq V$ be a subset of vertices with $\omega'(X) = t'$.

We define $r := |S \setminus X|$ and suppose that $r > k^4$. Since $|X \setminus S| \leq |V \setminus S| \leq k^3$, we get $\omega'(X \setminus S) \leq k^4$. It follows that

$$\begin{aligned}\omega'(X) &= \omega'(S) - \omega'(S \setminus X) + \omega'(X \setminus S)\\ &\leq i(|S| - r) + k^4\\ &< i(|S| - r) + ir\\ &= i|S|\\ &\leq t',\end{aligned}$$

a contradiction.

So, $r = |S \setminus X| \leq k^4$ and hence, together with $k^4 \leq |S_i|$ and $\omega'\big|_S \equiv i$, we may assume that $S \setminus X \subseteq S_i$. Otherwise we can exchange vertices of $(S \setminus X) \setminus S_i$ with vertices of $X \cap S_i$ until $S \setminus X \subseteq S_i$. In doing so, we maintain the total weight t' as well as the property whether being a minimal dominating set, by Lemma 4.3.19. This yields

$$\begin{aligned}\omega(X) &= \omega'(X) - \omega'(X \cap S) + \omega(X \cap S)\\ &= t' - i(|S| - r) + \omega(X \cap S)\\ &= t' - i|S| + (\omega(X \cap S) + ir)\\ &= t' - \omega'(S) + (\omega(X \cap S) + \omega(S \setminus X))\\ &= t' - \omega'(S) + \omega(S)\\ &= t.\end{aligned}$$

Hence, X is a minimal dominating set of G and the proof is finished. □

It follows that if we want to check a graph like the one given in Lemma 5.3.3 for k-equidomination, it is sufficient to work with k-equidominating functions that are constant on the isolated stable set class.

Now, we gathered together everything to prove Theorem 5.3.1.

Proof of Theorem 5.3.1. Let G be a graph with $|V(G)| = n$, $|E(G)| = m$ and $k \in \mathbb{N}$. First, we decompose G into pseudo classes. If there are more than k pseudo classes, then G is not k-equidominating, by Corollary 4.3.18. Otherwise, following Lemma 5.3.2, two cases may occur. If none of the cases are fulfilled, then G is also not k-equidominating.

Case 1. *Every pseudo class of size at least k^2, if any, is either a clique class, a clique bundle or a stable set bundle.*

As far as existent, we take one vertex of every singleton class, clique class and clique bundle, two vertices of an isolated stable set bundle, all vertices of an isolated stable set class and one neighbor of every other stable set class and stable set bundle. By this, we get a dominating set of size at most $k^2 + k$, which contains a minimal dominating set. This means $t \leq k^3 + k^2$ must hold for every k-equidominating structure (ω, t) of G. Then, we compute all mds-vectors and check if one of them has a component greater than $k^3 + k^2$. If so, G is not k-equidominating, by Lemma 5.1.5.

Otherwise, we perform the r-Clique Class Reduction rule and the r-Stable Set Bundle Reduction rule with $r = k^3 + k^2$ and $r = \lceil (k^3 + k^2)/2 \rceil$, respectively. Next, we compute the pseudo graph of the reduced graph and apply the r-Pseudo Graph Reduction rule with $r = k^3 + k^2$. We obtain a pseudo graph $\mathfrak{P}'' = (V'', \mathcal{P}'', \mu'')$ with $|V''| = \mathcal{O}(k^{3k+1})$ (compare the proof of Theorem 5.2.1). Again, by Lemma 5.1.2, Lemma 5.1.3 and Lemma 5.1.15, the obtained pseudo graph $\mathfrak{P}''$ is k-equidense if and only if G is k-equidominating.

Case 2. *There is a unique stable set class S with $|S| \geq k^2$ and every other pseudo class has fewer than k^2 vertices.*

In this case $|V(G) \setminus S| \leq k^3$, since there are at most $k - 1$ pseudo classes besides S and each of them has fewer than k^2 vertices. Further, we can assume that $|S| \geq k^5$ since otherwise $|V(G)| \leq k^5 + k^3 = \mathcal{O}(k^{3k+1})$ and the proof is finished. We distinguish two subcases.

Case 2.1. The stable set class S does not see any other twin class.
We now construct a graph G' by deleting all but k^5 many vertices of S and claim that G' is k-equidominating if and only if G is k-equidominating. Let $S' := S \cap V(G')$ be the set of the remaining vertices of S. Note that S' and S are contained in every minimal dominating set of G' and G, respectively.

$\underline{\Longleftarrow}$: Let G be k-equidominating and (ω, t) a k-equidominating structure. We show that (ω', t') with $\omega' := \omega\big|_{V(G')}$ and $t' := t - \omega(S \setminus S')$ is a k-equidominating structure of G'.

Let D' be a minimal dominating set of G'. Then, $D' \,\dot\cup\, (S \setminus S')$ is a minimal dominating set of G and consequently $\omega'(D') = \omega(D') = t - \omega(S \setminus S') = t'$.

Now, let $X' \subseteq V(G')$ be a subset of vertices with $\omega'(X') = t'$. Then, $\omega(X' \,\dot\cup\, (S \setminus S')) = t$ and hence $X' \,\dot\cup\, (S \setminus S')$ is a minimal dominating set of G. It follows that X' is a minimal dominating set of G'.

$\underline{\Longrightarrow}$: Let G' be k-equidominating with k-equidominating structure (ω', t'). Since all conditions of Lemma 5.3.3 are met, we may assume that $\omega'\big|_{S'} \equiv i$ for some $i \in [k]$. We prove that (ω, t) with

$$\omega(v) := \begin{cases} \omega'(v), & \text{if } v \in V(G'), \\ i, & \text{if } v \in S \setminus S', \end{cases}$$

and $t := t' + i|S \setminus S'|$ is a k-equidominating structure of G.

Let $D \subseteq V(G)$ be a minimal dominating set of G. Then there is a minimal dominating set $D' \subseteq V(G')$ of G' such that $D = (S \setminus S') \dot{\cup} D'$. Thus $\omega(D) = \omega(S \setminus S') + \omega(D') = i|S \setminus S'| + \omega'(D') = t$.

Now let $X \subseteq V(G)$ be a subset $\omega(X) = t$. By $|V(G) \setminus S| \leq k^3$, we get that $\omega(V(G) \setminus S) \leq k^4$. Suppose $|S \setminus X| > k^4$. Then

$$\begin{aligned}
\omega(X) &= \omega(X \cap S) + \omega(X \cap (V(G) \setminus S)) \\
&\leq \omega(S) - \omega(S \setminus X) + \omega(V(G) \setminus S) \\
&\leq i|S| - i|S \setminus X| + k^4 \\
&< i|S| - ik^4 + k^4 \\
&\leq i|S| \\
&= \omega'(S') + i|S \setminus S'| \\
&\leq t' + i|S \setminus S'| \\
&= t\,,
\end{aligned}$$

a contradiction. Thus, $|S \setminus X| \leq k^4$ and together with $k^4 \leq |S'|$ we may assume that $S \setminus X \subseteq S'$ (compare proof of Lemma 5.3.3). With $X' := X \cap V(G')$ we get $S \setminus S' = X \setminus X'$. It follows that $\omega'(X') = \omega(X) - i|X \setminus X'| = \omega(X) - i|S \setminus S'| = t'$. So, X' is a minimal dominating set of G' and consequently X is a minimal dominating set of G. Hence, G is k-equidominating and the claim is proved.

Taken together, we have proved that it is sufficient to check whether G' is k-equidominating with $|V(G')| \leq k^5 + k^3 = \mathcal{O}(k^{3k+1})$.

Case 2.2. The stable set class S sees another twin class T.
The idea to show that this subcase can not occur is the following: on the one hand, you can extend S to a minimal dominating set, which is of relatively large size. However, on the other hand – due to the non-empty neighborhood of S – there is also a minimal dominating set that does not intersect with S at all and therefore is relatively small.

So, suppose G is k-equidominating with k-equidominating structure (ω, t) and let D be a minimal dominating set with $D \cap S = \emptyset$ (compare Observation 4.3.9). It follows that $|D| \leq |V(G) \setminus S| \leq k^3$ and therefore $t \leq k^4$. But at the same time $t \geq \omega(S) \geq k^5$, a contradiction. Hence, G is not k-equidominating.

The determination of the pseudo class partition, the mds-vectors and the pseudo graph as well as the application of the reduction rules can be done in polynomial time (compare the proof of Theorem 5.2.1). This finishes the proof of the first assertion of this theorem.

Having the pseudo class partition and the mds-vectors at hand, we conclude that G is not k-equidominating if there are more than k pseudo classes, two stable set classes of size at least k^2 or a non-isolated stable set class of size at least k^5. If no stable set class of size at least k^2 exists, we first apply the three reduction rules in linear time and then Algorithm 2 to the obtained pseudo graph $\mathfrak{P}''$ of size $n'' = \mathcal{O}(k^{3k+1})$ in time

$$\mathcal{O}\left(n''^k k^k + n''^{2k+2} k^{-k-1} + k^{3k+3}\right) = \mathcal{O}\left(k^{3k^2+2k} + k^{6k^2+7k+1} + k^{3k+3}\right)$$
$$= \mathcal{O}\left(k^{6k^2+7k+1}\right) .$$

If an isolated stable set class exists, then we apply Algorithm 2 to the graph G' on at most $k^5 + k^3$ vertices, obtained by reducing the stable set class to k^5 vertices (if necessary).

This yields a total running time (compare proof of Theorem 5.2.1) of

$$\mathcal{O}\left(nm^2 + n^2 + k^{6k^2+7k+1}\right) .$$

□

In the proof of Theorem 5.3.1, more precisely in Case 1 of the proof, we obtain an upper bound of the target value of an equidominating structure by constructing a dominating set. By this, we make use of one of the few advantages of equidomination in comparison with equistability.

5.4 k-W-Equidomination

We refine the graph property k-equidominating by allowing k different weights that do not need to be of the set $[k]$. However, the weights have to be less than or equal to a given natural number, which will be a second parameter.

Definition 5.4.1. *For given $k, W \in \mathbb{N}$, a graph $G = (V, E)$ is said to be **k-W-equidominating** if there is an equidominating structure (ω, t) with $\omega\colon V \to \{w_1, \ldots, w_k\}$ such that $1 \le w_i \le W$ $(i = 1, \ldots, k)$ for some $t \in \mathbb{N}$. The pair (ω, t) is called a **k-W-equidominating structure** and ω a **k-W-equidominating function**.*

For a better understanding, let us take a look again at the star graphs $K_{1,n}$ $(n \in \mathbb{N})$ with one universal vertex v of degree n and n vertices $w_1, \ldots, w_n$ of degree 1. Every $K_{1,n}$ has exactly two minimal dominating sets, namely $\{v\}$ and $\{w_1, \ldots, w_n\}$. That means, in any k-W-equidominating function v has weight at least n and the sum of the weights of the w_i must equal the weight of v. However, one does not necessarily need many different weights. Indeed, two weights are sufficient, for example $\omega(v) = n$ and $\omega(w_i) = 1$, for $i \in [n]$, with target value $t = n$. Taken together, $K_{1,n}$ is 2-n-equidominating and n-equidominating. However, it is not n'-equidominating for any $n' < n$.

It is easy to see that every k-W-equidominating graph is W-equidominating, every k-equidominating graph is k-k-equidominating and every k-W-equidominating graph is k'-W'-equidominating for $k' \ge k$, $W' \ge W$. So, k-W-equidomination is a generalization of

k-equidomination. We started with k-equidomination, though, since we believe that this provides a better access to the topic. In the following we will always assume that $W > k$, since otherwise a graph G is k-W-equidominating if and only if G is W-equidominating and so the question whether a graph is k-W-equidominating would reduce to the already known case.

Analogously to the parameterized problems TARGET-t EQUIDOMINATION and k-EQUIDOMINATION, we define the following decision problem.

k-W-EQUIDOMINATION

Instance: A graph G and k, $W \in \mathbb{N}$.
Parameter: $k + W$.
Problem: Decide whether G is k-W-equidominating.

The results for k-equidomination as well as their proofs can be adapted to k-W-equidomination. The crucial element in doing so is considering for every k, that appears in the theorems, proofs and algorithms of Section 5.3, whether it appears due to the fact that k different weights are available or because of its actual value.

A good example for that is Lemma 5.3.2 on page 69. In this lemma, we show that there is at most one stable set class of size at least k^2 in a k-equidominating graph. In its proof, we exchange two subsets of identical total weight of two different stable set classes to obtain a contradiction. For this, it is necessary that a weight occurs in each stable set class at least as many times as the maximal possible weight. To ensure this, the stable set classes must contain more vertices than the number of different weights times the maximal weight value. Thus, in the k-W-equidominating case one of the appearing k in Lemma 5.3.2 remains and the other one is replaced by W. Hence, a graph G is not k-W-equidominating if G has two pseudo classes of size at least kW such that one of those pseudo classes is a stable set class.

For Algorithm 2 it is regardless which magnitude the weights have. The algorithm checks different possibilities of allocating the weights to the vertices. For this only the number of different possible weights is important. However, by Definition 5.4.1 the actual weights are not prescribed. Thus, we have to apply Algorithm 2 for every possible choice of k weights from the set $[W]$. Hence, we achieve the following result.

Corollary 5.4.2. *For given k, $W \in \mathbb{N}$, it is decidable whether a graph $G = (V, E)$ is k-W-equidominating or not in time $\mathcal{O}\left(nm^2 + \binom{W}{k}\left(n^k k^k + n^{2k+2}k^{-k-1} + k^{3k+3}\right)\right)$ ($n = |V|$ and $m = |E|$) and a k-W-equidominating structure is computed in this time.*

For the construction of a generalized kernel to obtain an FPT result, the situation is somewhat more complicated. However, the reduction rules can be used exactly in the same way as in the k-equidominating case. Proving that the Clique Class Reduction, the Stable Set Bundle Reduction and the Pseudo Graph Reduction rules are safe for k-W-equidomination can be done analogously to the proofs of Lemma 5.1.2, Lemma 5.1.3 and Lemma 5.1.15, respectively.

For proving the actual *FPT* result there is slightly more to do. As already discussed, the k^2 in Lemma 5.3.2 changes to kW. In Lemma 5.3.3, the isolated stable set class must be of size at least k^3W^2 and the rest of the graph of size at most k^2W. With that, we can prove the following corollary in the same way as Theorem 5.3.1.

Corollary 5.4.3. *The k-W-*Equidomination *problem admits a generalized kernel of size $\mathcal{O}\left(k^{k+1}W^{2k}\right)$ which is computable in polynomial time. Furthermore, there is an algorithm to solve the k-W-*Equidomination *problem for a given graph on n vertices and m edges which runs in time $\mathcal{O}\left(nm^2 + n^2 + \left(\binom{W}{k}\left(k^{2k^2+3k+1}W^{4k^2+4k}\right)\right)\right)$.*

Chapter 6

Conclusion and Outlook

In Section 6.1, we give a summary of our research results. Then, we state several detailed conjectures and assertions that might be promising and helpful for further investigations on equidomination in Section 6.2. We end the chapter by pointing out possible directions of future research in Section 6.3.

6.1 Our Contribution

In this thesis we dedicated ourselves to the research on equidomination. The initial motivation was to obtain fixed-parameter tractability results for two parameterized versions of the EQUIDOMINATION problem. This turned out to be significantly harder than we expected. However, on the way we achieved some fruitful results using interesting techniques.

Since hardly any results regarding equidomination existed, we started by collecting various general results: we proved that one cannot characterize the class of equidominating graphs in terms of forbidden induced subgraphs. For several standard graphs – K_n, $\overline{K_n}$, $K_{n,m}$, $T(2n, n)$, P_n and C_n (n, $m \in \mathbb{N}$) – we examined whether or not they are equidominating.

Furthermore, we investigated to what extent certain operations are compatible with equidomination. Adding and deleting isolated and universal vertices preserve the property of being equidominating or not. We showed that arbitrarily connecting the universal vertices of two (non-complete) equidominating graphs yields an equidominating graph. Consequently, the disjoint union and the chain-join preserve equidomination.

Next, we focused on the class of hereditarily equidominating graphs. We showed that a graph is hereditarily equidominating if and only if none of its induced subgraphs is isomorphic to the graphs P_5, C_5, *bull*, *banner*, *house*, $K_{2,3}$ or $\overline{P_2 \cup P_3}$ (see Figure 3.2 on page 26). Further, we gave a structural decomposition: every hereditarily equidominating graph is either K_1 or $T(2n, n)$, for $n \in \mathbb{N}$. Or it can be obtained by adding a universal vertex to a hereditarily equidominating graph or by a chain-join of two hereditarily equidominating graphs. We gave an explicit recognition algorithm based on this decomposition, with a running time of $\mathcal{O}(n(n + m))$ for a graph on n vertices and m

edges. Using that the prime induced subgraphs of hereditarily equidominating graphs are of bounded clique-width, we proved the existence of a linear time recognition algorithm applying a meta-theorem of Courcelle.

Then, we turned our attention to complexity issues. We introduced two parameterized versions of the EQUIDOMINATION problem: the k-EQUIDOMINATION problem (allow only vertex weights up to a certain value $k \in \mathbb{N}$) and the TARGET-t EQUIDOMINATION problem (predetermine the target value $t \in \mathbb{N}$). In order to prove that these problems lie in the complexity class FPT, we needed a partition of the vertices of a graph such that vertices of different blocks must have different weights with respect to any equidominating function. As the twin partition seemed to be promising but not completely suffices the desired condition, we examined in which cases the vertices of different twin classes can have the same weight. It turned out that this can happen only if such vertices are adjacent.

Our examination led us to a coarsening of the twin partition: the so-called pseudo class partition. Each pseudo class is either a twin class, a maximal collection of adjacent, mds-exchangeable clique classes or a maximal collection of adjacent stable set classes of size two with identical neighborhoods. Conveniently, every graph has a unique pseudo class partition. Using the pseudo class partition, we stated an XP algorithm that decides if a graph on n vertices and m edges is k-equidominating in time $\mathcal{O}\left(nm^2 + n^k k^k + n^{2k+2}k^{-k-1} + k^{3k+3}\right)$. The algorithm can easily be modified to also solve the TARGET-t EQUIDOMINATION problem. In this case, for $t \in \mathbb{N}$, we achieve a running time of $\mathcal{O}\left(nm^2 + n^t t^t + n^{2t+2}t^{-t-1} + t^{3t+3}\right)$.

The pseudo class partition is one of the centerpieces to obtain the fixed-parameter tractability results. The other centerpiece is the examination of three reduction rules, each of them dealing with a particular pseudo class. The first one is the r-Clique Class Reduction and the second one the r-Stable Set Bundle Reduction. Having an upper bound r on the potential target value, we proved that one can safely reduce the number of vertices of any clique class and stable set bundle to r. This means that the original graphs is target-t k-equidominating if and only if the reduced graph is target-t k-equidominating.

The third reduction rule, the r-Pseudo Graph Reduction, concerns clique bundles. Proving that we can use it for our purposes required a little more work. We introduced pseudo graphs, a mathematical object to which we transformed all relevant information (with respect to equidomination) of a graph. With this somewhat technical workaround we were able to reduce the size of clique bundles, the last piece of the puzzle.

We proved that the TARGET-t EQUIDOMINATION problem is fixed-parameter tractable. Using the three above-mentioned reduction rules, we obtained a generalized kernel of size $\mathcal{O}\left(t^{t+1}\right)$ in polynomial time. Furthermore, by applying the XP algorithm to the kernel, we achieved an FPT algorithm which runs for a graph on n vertices and m edges in time $\mathcal{O}(nm^2 + n^2 + t^{2t^2+3t+1})$.

For the k-Equidomination problem there was slightly more to do. This is because – in contrast to a target-t equidominating graph – stable set classes can have arbitrary many vertices in a k-equidominating graphs. Nevertheless, we overcame this difficulty and showed that the k-Equidomination problem lies in FPT, too. Again, we constructed a generalized kernel in polynomial time. The kernel of the k-Equidomination problem is of size $\mathcal{O}(k^{3k+1})$ and the FPT algorithm needs time $\mathcal{O}(nm^2 + n^2 + k^{6k^2+7k+1})$.

Moreover, we generalized the k-Equidomination problem by allowing k different weights that do not need to be of the set $[k]$, but less than or equal to a second parameter $W > k$. For the resulting k-W-Equidomination problem we proved analogous results. It admits a kernel of size $\mathcal{O}\left(k^{k+1}W^{2k}\right)$, to which we can also apply the XP algorithm. Since we do not prescribe the weights, we achieve an FPT algorithm with running time $\mathcal{O}\Big(nm^2 + n^2 + \big(\binom{W}{k}(k^{2k^2+3k+1}W^{4k^2+4k})\big)\Big)$ for the k-W-Equidomination problem.

6.2 Discussion

In this section, we state several rather detailed and technical conjectures and assertions that came up during our studies. A wider outlook to future research is given in Section 6.3. For the reader's convenience and a better overview, we present the questions and assertions in a list. Afterwards, we discuss and explain some points more detailed.

Since in the end clique bundles – more precisely, clique bundles that contain only singleton classes – gave us the most headaches and since the sizes of the kernels are mostly depending on them, a lot of the following concerns clique bundles. Most of the questions should be also asked explicitly for equidominating graphs if they do not hold in general.

In the following, let P_i and P_j be clique bundles of a graph. For the sake of clarity, we omit a complete declaration of the used variables here. The reader is referred to Chapter 4 and Chapter 5 for full definitions, which may lead to a better understanding of the upcoming list:

(1) Let D be a minimal dominating set with $|D \cap P_i| = 1$. Then, each private neighbor of a vertex of $D \cap P_i$ either lies in P_i or is adjacent to every vertex of P_i.

(2) If two vertices v, $w \in P_i$ exists with $N[v] \subseteq N[w]$, then $|D \cap P_i| \leq 1$ holds for every minimal dominating set D.

(3) If $|D \cap P_i| \leq 1$ for all minimal dominating sets D, then we can apply a reduction rule that deletes particular edges incident to the vertices of P_i (see Lemma A.1 in Appendix A).

(4) If $|D \cap P_i| \leq 1$ and $|D \cap P_j| \leq 1$ for all minimal dominating sets D, then we can apply a reduction rule that deletes particular edges incident to vertices of both P_i and P_j (see Lemma A.2 in Appendix A).

(5) What are necessary and sufficient conditions for $|D \cap P_i| \leq 1$ for all minimal dominating sets D?

(6) Even though there might be a minimal dominating set containing numerous vertices of a clique bundle, there is always also a minimal dominating set that contains exactly one vertex of this clique bundle.

(7) Let D be a minimal dominating set with $l := |D \cap P_i| > 1$. Further, let $P' \subseteq P_i$ be an arbitrary subset of P_i with $|P'| = m$. Then, $(D \setminus P_i) \cup P'$ is a minimal dominating, a dominating but not minimal or a non-dominating set if $m = l$, $m > l$ or $m < l$, respectively.

(8) Let v, $w \in P_i$ and let x be a vertex with $\mu_i^x > 0$ that is adjacent to v and non-adjacent to w. Then, x has a neighbor y that is non-adjacent to v and adjacent to w with $N(y) \setminus \{w\} \subseteq N[x]$.

(9) Let $v \notin P_i$. Is there always a minimal dominating set D with $|D \cap P_i| = \mu_i^v$?

(10) Is it possible that $\mu_i^v \notin \{0, 1, 2, |P_i|\}$ for a vertex $v \notin P_i$?

(11) Is there always an equidominating function that is constant on a stable set class? If so, it there also one that is constant on all stable set classes?

(12) Is it possible that a minimal dominating set of an equidominating graph contains exactly one vertex of a stable set class or a stable set bundle?

The assertions (1)-(4) concern clique bundles that intersects with minimal dominating sets in at most one vertex. For a clique bundle P_i, this is the case if and only if $\mu_i^v \leq 1$ holds for every vertex v, which we can check in polynomial time. If this holds for every clique bundle of a graph, then we can use the reduction rules of (3) and (4) to obtain smaller problem kernels. In particular, one should use the reduction rules for an implementation of one of the discussed algorithms.

The assertions (6) and (7) might be usefull to answer (5). Assertion (8) results from the opposing effects of mds-exchangeability and the definition of the mds-vectors with respect to a clique bundle: since v and w are mds-exchangeable, x has a neighbor y that only sees neighbors of x, and possibly w (compare Figure 4.4 and Algorithm 1 on page 50). But at the same time there exists a minimal dominating set such that x is only dominated by vertices of P_i (since $\mu_i^x > 0$). This means y must be adjacent to w.

Regarding (10), the interesting cases are $\mu_i^v > 1$. However, (besides 0 and 1) we only obtained 2 and $|P_i|$ as values for μ_i^v during our studies (see for example the graph shown in Figure 5.1 on page 61). More specific, in every example we worked with, the following situation occurred: every vertex of the clique bundle has some kind of partner outside and is the only one that either does or does not see his partner. For such a partner $v \notin P_i$. we obtain $\mu_i^v = |P_i|$ in the first case and $\mu_i^v = 1$ in the latter case.

The graph of Figure 4.5(a) on page 52 shows that (12) can be answered with yes in general. But it might be impossible in an equidominating graph.

6.3 Future Directions

During the studies on equidomination, several questions and ideas arose that provide a base for future research.

Most importantly, we would like to determine the complexity class the EQUIDOMINATION problem lies in. We believe that it is at least NP-hard. It also seems interesting to look out for complexity results for the EQUIDOMINATION problem when the input is restricted to a certain graph class. Moreover, can one deduce complexity results for well-studied decision problems on the class of equidominating graphs?

One could take further parameterizations into account, for example, using the tree-width or the clique-width of a graph as parameter. Another frequently used parameter in the field of parameterized complexity is the vertex cover number. There are direct relations between the vertex cover number and the clique-width on the one hand and the number of twin classes and hence of pseudo classes on the other hand (compare [41]). It seems that the so-called twin-cover [30] is also related to the topic equidomination. Is it possible to deduce FPT results when the EQUIDOMINATION problem is parameterized by one of those graph parameters?

Regarding our parameterizations, it is unanswered if we can further reduce the size of the kernels, if it is possible to achieve kernels of polynomial size (in terms of the parameter) or if no of such sized kernels exist at all. Since the sizes of the obtained kernels are determined by clique bundles, one should begin by trying to achieve a stronger reduction rule for clique bundles.

For all of the previously mentioned questions, however, it might be necessary to get a better grip on the combinatorial properties of equidominating graphs. A first step in this direction could be a characterization of target-t equidominating and k-equidominating graphs for small t and k. We already know that complete graphs are the only target-1 equidominating graphs and that a 1-equidominating graph is either a complete, an edgeless or a particular Turán graph (namely $T(2n, n)$ for $n \in \mathbb{N}$). Using Corollary 4.3.18 on page 48, it is possible to describe the corresponding classes for small t, $k > 1$.

We believe that the most promising access to a better understanding for equidominating graphs and to achieve further results is to intensify analyzing clique bundles, the relationship between them and to other pseudo classes. In the previous section, we already stated several ideas how to approach this issue.

Furthermore, it seems reasonable to search for more operations on graphs that preserve equidomination (compare Chapter 3). Can we characterize equidominating graphs by such operations like we did for hereditarily equidominating graphs in Theorem 3.3.2?

There exists an inclusion chain for several super- and subclasses of the class of equistable graphs (see [48]). Which graph classes are contained in the class of equidominating graphs and which graph classes contain this class? It is already known that no inclusion relation exists between the class of equidominating graphs on the one hand and domishold or equistable graphs on the other hand (see [44] and [56]). On the affirmative side, the class of threshold graphs is contained the class of equidominating graphs.

In general, equistability and related topics provide an excellent pattern for new research directions. For example, one could define and investigate domination related properties analogous to interstable (taking a target interval instead of a target value) or d-equistable (using vectorial weights and target values), both defined in [50]. Moreover, one could adapt the idea of characterizing minimal dominating sets by a weight function and a target or threshold value to other forms of domination like total, multiple, global and efficient domination (see [35] for definitions). This concept can also be extended to any property on subsets of vertices or even on subsets of edges, for example, being a (maximal) matching. Thoughts in this direction could concern a characterization of the resulting graph classes as well as the (parameterized) complexity of the according decision problems.

Furthermore, Chiarelli and Milanič discovered an interesting relation between so-called total domishold graphs and Boolean functions [14]. Are Boolean functions also useful with respect to equidomination?

Finally, one could consider exploring further applications of the pseudo class partition and pseudo graphs (in particular when induced by a graph).

We hope that one day equidomination will gain as many attention as equistability such that more results will appear. This thesis could be the first step in this direction.

Appendix A

Further Reduction Rules

The following two rules can be applied to clique bundles $\mathcal{C}$ for which $|D \cap \mathcal{C}| \leq 1$ holds for every minimal dominating set D. Let us call this **Condition A**. This is the case, for example, if a clique bundle contains at least one clique class (and not only singleton classes). More general, if there are two vertices v, $w \in \mathcal{C}$ with $N[v] \subseteq N[w]$. However, Condition A might also be fulfilled in other situations.

The first reduction rule concerns clique bundles that contain two twin classes where the neighborhood of one twin class is contained in the neighborhood of the other one. Note that we talk about twin classes here since it could be either a clique class or a singleton class. Since two such twin classes exist, Condition A is met.

> **Clique Bundle Neighborhood Reduction**: If there are two twin classes T_1 and T_2 in a clique bundle with $N[T_1] \subseteq N[T_2]$, delete all edges between the vertices of T_2 and $N[T_2] \setminus N[T_1]$.

Lemma A.1. *Let $G = (V, E)$ be a graph with a clique bundle $\mathcal{C}$ and let $T_1, T_2 \subseteq \mathcal{C}$ be two twin classes with $N[T_1] \subseteq N[T_2]$. Let G' be the graph obtained from G by using the Clique Bundle Neighborhood Reduction rule with respect to T_1 and T_2, that is*

$$G' = \Big(V, E \setminus \big\{vw \in E \mid v \in T_2, w \in N[T_2] \setminus N[T_1]\big\}\Big).$$

Let $D \subseteq V$. Then D is a minimal dominating set of G if and only if D is a minimal dominating set of G'.

Proof. $\Longrightarrow$: Let $D \subseteq V$ be a minimal dominating set of G and suppose that D is not dominating in G'. Then either a vertex $v \in T_2$ or a vertex $w \in N[T_2] \setminus N[T_1]$ is not dominated. In the first case, T_1 is also not dominated, neither in G' nor in G, a contradiction. In the latter case, w is a private neighbor of T_2 not seeing T_1. If we now exchange the (unique) vertex of $T_2 \cap D$ with a vertex of T_1, then w is not longer dominated, again a contradiction. Thus, D is a dominating set of G' and since we only deleted edges of G to obtain G', D is a minimal dominating set of G'.

$\Longleftarrow$: Let $D \subseteq V$ be a minimal dominating set of G'. Since $V(G') = V(G)$ and $E(G') \subseteq E(G)$, we get $N_G[D] \supseteq N_{G'}[D] = V$. Thus, D is also a dominating set of G. Suppose that D is not minimal in G. Then there exists a minimal dominating set $\widetilde{D} \subsetneq D$ of G. However, according to the first part of the proof, $\widetilde{D}$ is a minimal dominating set of G', a contradiction. □

So, when applying the Clique Bundle Neighborhood Reduction rule, we join two clique classes to only one. This implies the following: if there is a twin class in a clique bundle which only sees the other twin classes of that clique bundle (see for example Figure 4.3), then, by applying the Clique Bundle Neighborhood Reduction several times, the clique bundle becomes a single, isolated clique class.

With the second reduction rule we reduce the edges between two clique bundles.

> **Clique Bundles Correlation Reduction**: If there are two adjacent twin classes $T_1 \in \mathcal{C}_1$ and $T_2 \in \mathcal{C}_2$ of two clique bundles $\mathcal{C}_1$ and $\mathcal{C}_2$ such that T_1 does not see every twin class of $\mathcal{C}_2$ and T_2 does not see every twin class of $\mathcal{C}_1$, then delete all edges between the vertices of T_1 and T_2.

Lemma A.2. *Let $G = (V, E)$ be a graph with two clique bundles $\mathcal{C}_1$ and $\mathcal{C}_2$ fulfilling Condition A. Let $T_1 \in \mathcal{C}_1$ and $T_2 \in \mathcal{C}_2$ be two adjacent twin classes with $\mathcal{C}_2 \nsubseteq N(T_1)$ and $\mathcal{C}_1 \nsubseteq N(T_2)$. Further, let G' be the graph obtained from G by applying the Clique Bundles Correlation Reduction rule with respect to T_1 and T_2, that is*

$$G' = \Big(V, E \setminus \big\{vw \in E \mid v \in T_1, w \in T_2\big\}\Big) .$$

Let $D \subseteq V$. Then D is a minimal dominating set of G if and only if D is a minimal dominating set of G'.

Proof. Analogously to the proof of Lemma A.1, it is sufficient to show that every minimal dominating set of G is also a dominating set of G'. So, let D be a minimal dominating set of G which is not dominating in G'. Then either T_1 or T_2 is not dominated. Without loss of generality let T_1 not be dominated in G'. That means there is a vertex $x \in D \cap T_2$ with T_1 as private neighbors, formally $T_1 \subseteq pn_G(x, D)$. However, if we now exchange x with a vertex of a twin class of $\mathcal{C}_2$ that is not seen by T_1 (in G), then T_1 is not dominated (in G). This contradicts the mds-exchangeable property of $\mathcal{C}_2$. □

It can be shown straightforwardly that the following observation holds. Together with Lemma A.1 and Lemma A.2, we obtain Corollary A.4.

Observation A.3. *Let k, $t \in \mathbb{N}$. If two graphs G and G' on the same vertex set (that is $V(G) = V(G')$) have exactly the same family of minimal dominating sets, then G is target-t k-equidominating if and only if G' is target-t k-equidominating.*

Corollary A.4. *In the context of Lemma A.1 or Lemma A.2, G is target-t k-equidominating if and only if G' is target-t k-equidominating.*

List of Figures

Bibliography

[1] Liliana Alcón, Marisa Gutierrez, István Kovács, Martin Milanič, and Romeo Rizzi. Strong cliques and equistability of EPT graphs. Discrete Applied Mathematics, 203:13–25, 2016.

[2] Noga Alon, Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving MAX-r-SAT above a tight lower bound. Algorithmica, 61(3):638–655, Nov 2011.

[3] Claude Benzaken and Peter L. Hammer. Linear separation of dominating sets in graphs. In Béla Bollobás, editor, Advances in Graph Theory, volume 3 of Annals of Discrete Mathematics, pages 1–10. Elsevier, 1978.

[4] Claude Berge. The theory of graphs. Courier Corporation, 1962.

[5] Peter Van Emde Boas. Machine models and simulations. Handbook of Theoretical Computer Science, volume A, pages 1–66, 1990.

[6] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. Journal of Computer and System Sciences, 75(8):423–434, 2009.

[7] Joseph Bonneau, Jonathan Anderson, Ross Anderson, and Frank Stajano. Eight friends are enough: Social graph approximation via public listings. In Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, SNS '09, pages 13–18. ACM New York, 2009.

[8] Endre Boros, Vladimir Gurvich, and Martin Milanič. On equistable, split, cis, and related classes of graphs. Discrete Applied Mathematics, 216(1):47–66, 2017.

[9] Andreas Brandstädt, Feodor F. Dragan, Hoàng-Oanh Le, and Raffaele Mosca. New graph classes of bounded clique-width. Theory of Computing Systems, 38(5):623–645, 2005.

[10] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. Graph classes: a survey. SIAM, 1999.

[11] Andreas Brandstädt, Arne Leitert, and Dieter Rautenbach. Efficient Dominating

and Edge Dominating Sets for Graphs and Hypergraphs, pages 267–277. Springer-Verlag Berlin Heidelberg, 2012.

[12] Eglantine Camby and Oliver Schaudt. The price of connectivity for dominating set: Upper bounds and complexity. Discrete Applied Mathematics, 177:53–59, 2014.

[13] Nina Chiarelli and Martin Milanič. Linear Separation of Total Dominating Sets in Graphs, pages 165–176. Springer-Verlag Berlin Heidelberg, 2013.

[14] Nina Chiarelli and Martin Milanič. Total domishold graphs: A generalization of threshold graphs, with connections to threshold hypergraphs. Discrete Applied Mathematics, 179:1–12, 2014.

[15] Nina Chiarelli and Martin Milanič. On a class of graphs between threshold and total domishold graphs. Discrete Applied Mathematics, 195:43–58, 2015.

[16] Václav Chvátal. On certain polytopes associated with graphs. Journal of Combinatorial Theory, Series B, 18(2):138–154, 1975.

[17] Václav Chvátal and Peter L. Hammer. Aggregation of inequalities in integer programming. Annals of Discrete Mathematics, 1:145–162, 1977.

[18] Bruno Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. Handbook of Graph Grammars and Computing by Graph Transformation, 1:313–400, 1997.

[19] Bruno Courcelle and Joost Engelfriet. Graph structure and monadic second-order logic. A language-theoretic approach. Encyclopedia of Mathematics and its applications, Vol. 138. Cambridge University Press, 2012.

[20] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. Theory of Computing Systems, 33(2):125–150, 2000.

[21] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. Discrete Applied Mathematics, 101(1):77–114, 2000.

[22] Alain Cournier and Michel Habib. A new linear algorithm for modular decomposition. In Sophie Tison, editor, Trees in Algebra and Programming CAAP'94, volume 787 of Lecture Notes in Computer Science, pages 68–84. Springer-Verlag Berlin Heidelberg, 1994.

[23] Reinhard Diestel. Graph Theory. Springer-Verlag, New York, 2016.

[24] Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and com-

pleteness. Congressus Numerantium, 87:161–187, 1992.

[25] Rod G. Downey and Michael R. Fellows. Fundamentals of Parameterized Complexity. Springer-Verlag Berlin Heidelberg, 2013.

[26] Paul Erdös. Problems and results in additive number theory. Colloque Theorie des Nombres, pages 127–137, 1956.

[27] Arthur Finbow, Bert Hartnell, and Richard Nowakowski. Well-dominated graphs: a collection of well-covered ones. Ars Combinatoria, 25A:5–10, 1988.

[28] Jörg Flum and Martin Grohe. Parameterized Complexity Theory. Springer-Verlag Berlin Heidelberg, 2006.

[29] Jean-Luc Fouquet. A decomposition for a class of (P_5, $\overline{P_5}$)-free graphs. Discrete Mathematics, 121(1-3):75–83, 1993.

[30] Robert Ganian. Twin-Cover: Beyond Vertex Cover in Parameterized Algorithmics, pages 259–271. Springer-Verlag Berlin Heidelberg, 2012.

[31] Michael R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, 1979.

[32] Gregory Gutin and Vadim E. Zverovich. Upper domination and upper irredundance perfect graphs. Discrete Mathematics, 190(1):95–105, 1998.

[33] Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. Computer Science Review, 4(1):41–59, 2010.

[34] Pinar Heggernes and Dieter Kratsch. Linear-time certifying recognition algorithms and forbidden induced subgraphs. Nordic Journal of Computing, 14(1-2):87–108, 2007.

[35] Teresa W. Heyes, Stephan T. Hedetniemi, and Peter J. Slater. Fundamentals of Domination in Graphs. Marcel Dekker Inc., 1998.

[36] Eun Jung Kim, Martin Milanič, and Oliver Schaudt. Recognizing k-equistable graphs in FPT time. 41st International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2015, pages 487–498, 2016.

[37] Ephraim Korach and Uri N. Peled. Equistable series-parallel graphs. Discrete Applied Mathematics, 132(1-3):149–162, 2003.

[38] Ephraim Korach, Uri N. Peled, and Udi Rotics. Equistable distance-hereditary graphs. Discrete Applied Mathematics, 156(4):462–477, 2008.

[39] Bernhard Korte and Jens Vygen. Combinatorial Optimization: Theory and Algorithms. Springer-Verlag Berlin Heidelberg, 2000.

[40] Yung-Ling Lai and Kenneth Williams. A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs. Journal of Graph Theory, 31(2):75–94, 1999.

[41] Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. Algorithmica, 64(1):19–37, Sep 2012.

[42] Vadim E. Levit and Martin Milanič. Equistable simplicial, very well-covered, and line graphs. Discrete Applied Mathematics, 165:205–212, 2014.

[43] Vadim E. Levit, Martin Milanič, and David Tankus. On the recognition of k-equistable graphs. In Martin C. Golumbic, Michal Stern, Avivit Levy, and Gila Morgenstern, editors, Graph-Theoretic Concepts in Computer Science, volume 7551 of Lecture Notes in Computer Science, pages 286–296. Springer-Verlag Berlin Heidelberg, 2012.

[44] Nadimpalli V. R. Mahadev and Uri N. Peled. Threshold graphs and related topics, volume 56 of Annals of Discrete Mathematics. North-Holland Publishing Co., Amsterdam, 1995.

[45] Nadimpalli V. R. Mahadev, Uri N. Peled, and Feng Sun. Equistable graphs. Journal of Graph Theory, 18(3):281–299, 1994.

[46] Paola Marchioro and Aurora Morgana. Structure and recognition of domishold graphs. Discrete Mathematics, 50:239–251, 1984.

[47] Ross M. McConnell and Jeremy P. Spinrad. Modular decomposition and transitive orientation. Discrete Mathematics, 201(13):189–241, 1999.

[48] Štefko Miklavič and Martin Milanič. Equistable graphs, general partition graphs, triangle graphs, and graph products. Discrete Applied Mathematics, 159(11):1148–1159, 2011.

[49] Martin Milanič, James Orlin, and Gábor Rudolf. Complexity results for equistable graphs and related classes. Annals of Operations Research, 188:359–370, 2011.

[50] Martin Milanič and Gábor Rudolf. Structural results for equistable graphs and related graph classes. RUTCOR Research Report 25-2009, 2009.

[51] Martin Milanič and Nicolas Trotignon. Equistarable graphs and counterexamples to three conjectures on equistable graphs. Journal of Graph Theory, 84(4):536–551, 2017.

[52] Rolf H. Möhring and Franz J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. In Rainer E. Burkard, Ray A. Cuninghame-Green, and Uwe Zimmermann, editors, Algebraic and Combinatorial Methods in Operations Research, volume 95 of North-Holland Mathematics Studies, pages 257–355. North-Holland, 1984.

[53] Raffaele Mosca. Polynomial algorithms for the maximum stable set problem on particular classes of P_5-free graphs. Information Processing Letters, 61(3):137–143, 1997.

[54] Oystein Ore. Theory of graphs. American Mathematical Society Providence, Rhode Island, 1962.

[55] Christos H. Papadimitriou and Kenneth Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Courier Corporation, 1982.

[56] Charles Payan. A class of threshold and domishold graphs: equistable and equidominating graphs. Discrete Mathematics, 29(1):47–52, 1980.

[57] Uri N. Peled and Udi Rotics. Equistable chordal graphs. Discrete Applied Mathematics, 132(1-3):203–210, 2003.

[58] Dieter Rautenbach and Vadim E. Zverovich. Perfect graphs of strong domination and independent strong domination. Discrete Mathematics, 226(1):297–311, 2001.

[59] Marc Tedder, Derek Corneil, Michel Habib, and Christophe Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, Automata, Languages and Programming, volume 5125 of Lecture Notes in Computer Science, pages 634–645. Springer-Verlag Berlin Heidelberg, 2008.

[60] Peter van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. In Report No. 81-04. Department of Mathematics, University of Amsterdam, 1981.

[61] Igor E. Zverovich and Vadim E. Zverovich. A characterization of domination perfect graphs. Journal of Graph Theory, 15(2):109–114, 1991.

Index

Page numbers in **bold** refer to definitions or first appearances. Variables or terms with a variable at the beginning are included in the alphabetical order, where symbols appear at the beginning of the index. The · symbolizes an argument of an operator.

Danksagung

Allen voran danke ich Prof. Dr. Rainer Schrader. Er schuf ein angenehmes wie familiäres Arbeitsklima und -umfeld. Bei Fragen und Angelegenheiten aller Art war er jederzeit nahezu unverzüglich mit Rat und Tat zur Stelle. Damit gab er mir beste Voraussetzungen für eine erfolgreiche Promotion.

Weiter danke ich Prof. Dr. Oliver Schaudt. Unsere intensiven Sessions waren ebenso ertragreich wie spaßbringend. Zweiteres mag an den unterhaltsamen Pausenthemen wie Gesellschaftsspiel- und Musikdiskussionen, Retrogames, Rätselraten sowie MMA und Ultimate liegen. Aber auch das Eintauchen in die Tiefen der Abstraktheit, ich sag nur Ballons oder Graphenüberlappungen, bereitete neben rauchenden Köpfen gleichermaßen viel Freunde.

Zudem danke ich meinen Arbeitsskollegen, die mich auf meinem Weg begleitet haben – oder wie ich gerne sage, Arbeitsfreunde. Zunächst sei die Bauspargruppe erwähnt: Dr. Thomas Chevalier, Marcel Schwalb, Dr. Dominique Ziegelmayer, Dr. Andrea Oversberg, Jun-Gyu Kim und Roland Mainka. Ich habe die Zusammenarbeit sehr genossen und für alle Lebenslagen und -fragen gab es den richtigen Ansprechpartner. In den letzten Jahren erlebten wir gemeinsam zu vieles, um hier einzelne Themen, Ereignisse und Albernheiten hervorzuheben. Aber auch jenseits des Flures, hatte ich das Privileg toller Kollegen. Ich danke meiner M. Dr. Vera Weil, die mir insbesondere am Ende meiner Promotion mit ihrer Erfahrung und Weisheit zur Seite stand. Alexander Apke und Toni Böhnlein danke ich für allerlei Diskussionen und anregende Donnerstagsseminare, vorzugsweise natürlich nicht am Donnerstag, sowie das kritische Lesen meiner Arbeit. Auch wenn sie am Ende gewisse Ausdrücke und Bilder nicht mehr hören und sehen konnten, hat man es ihnen kaum angemerkt. Dank ihnen allen fühlte es sich in mancher Hinsicht gar nicht so an, als würde man zur Arbeit gehen. Es ist sehr viel wert, jeden Tag auch Spaß zu haben und zu lachen.

Ein herzlicher Dank geht weiterhin an Annette Koenen für schnelle und unkomplizierte Hilfe in allen Belangen sowie an Martin Olschewski, der immer mit technischer Beratung zu Stelle war und mir immer öfters gelöschte oder überschriebene Dateien wiederherstellte.

Ein besonders großer Dank gilt meinen Eltern, die mich mein Leben lang unterstützt, gefördert und gefordert haben. Damit haben sie letztlich das Fundament für diese Arbeit gelegt.

Explizit danke ich meinen (bisher nicht erwähnten) Korrekturlesern: Berna Gülmez, Marvin Brandau, Patrick Senger und Yan Hoepfner! Unabhängig davon danke ich allen Freunden und lieben Menschen, denen ich bisher begegnen durfte. Ohne euch wäre ich nicht der Mensch, der ich heute bin!

Katja, ich bin sehr froh, dass es Dich gibt und noch froher, dass sich unsere Wege gekreuzt haben! Ich danke Dir für deine Unterstützung, insbesondere in den letzten sieben Monaten.

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzen Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie – abgesehen von unten angegebenen Teilpublikationen – noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss der Promotionsverfahrens nicht vornehmen werde.

Die Bestimmung der Promotionsordnung sind mit bekannt. Die von mir vorgelegte Dissertation ist von Herrn Prof. Dr. Rainer Schrader betreut worden.

Fabian Senger

Teilpublikationen

- Oliver Schaudt and Fabian Senger. The Parameterized Complexity of the Equidomination Problem. In Hans L. Bodlaender and Gerhard J. Woeginger, editors, Graph-Theoretic Concepts in Computer Science: 43nd International Workshop, WG 2017, Eindhoven, The Netherlands, June 21-23, 2017, Revised Selected Papers, volume 10520 of Theoretical Computer Science and General Issues. Springer-Verlag Berlin Heidelberg, 2017.