

Residue Domination in Bounded-Treewidth Graphs

Jakob Greilhuber  

TU Wien, Austria

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Philipp Schepper 

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Philip Wellnitz  

National Institute of Informatics, Tokyo, Japan

The Graduate University for Advanced Studies, SOKENDAI, Tokyo, Japan

Abstract

For the vertex selection problem (σ, ρ) -DOMSET one is given two fixed sets σ and ρ of integers and the task is to decide whether we can select vertices of the input graph such that, for every selected vertex, the number of selected neighbors is in σ and, for every unselected vertex, the number of selected neighbors is in ρ [Telle, Nord. J. Comp. 1994]. This framework covers many fundamental graph problems such as INDEPENDENT SET and DOMINATING SET.

We significantly extend the recent result by Focke et al. [SODA 2023] to investigate the case when σ and ρ are two (potentially different) residue classes modulo $m \geq 2$. We study the problem parameterized by treewidth and present an algorithm that solves in time $m^{\text{tw}} \cdot n^{\mathcal{O}(1)}$ the decision, minimization and maximization version of the problem. This significantly improves upon the known algorithms where for the case $m \geq 3$ not even an explicit running time is known. We complement our algorithm by providing matching lower bounds which state that there is no $(m - \varepsilon)^{\text{pw}} \cdot n^{\mathcal{O}(1)}$ -time algorithm parameterized by pathwidth pw , unless SETH fails. For $m = 2$, we extend these bounds to the minimization version as the decision version is efficiently solvable.

2012 ACM Subject Classification Theory of computation \rightarrow Parameterized complexity and exact algorithms

Keywords and phrases Parameterized Complexity, Treewidth, Generalized Dominating Set, Strong Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.41

Related Version The Master's Thesis of Jakob Greilhuber [36] is based on the lower bound results of this work.

Full Version: <https://arxiv.org/abs/2403.07524>

Funding *Jakob Greilhuber:* Part of Saarbrücken Graduate School of Computer Science, Germany.

Philipp Schepper: Part of Saarbrücken Graduate School of Computer Science, Germany.

Acknowledgements The work of Jakob Greilhuber has been carried out mostly during a summer internship at the Max Planck Institute for Informatics, Saarbrücken, Germany. The work of Philip Wellnitz was partially carried out at the Max Planck Institute for Informatics.

1 Introduction

Classical graph problems such as Dominating Set or Independent Set are ubiquitous in computer science. These problems are not only of theoretical interest but also have many practical applications; including facility location, coding theory, modeling communication networks, map labeling, or even similarity measures on molecules [3, 4, 16, 31, 38, 39]. Therefore, these problems are extensively studied on plenty of graph classes and several generalizations and variations have been formulated and considered [7, 12, 22, 30, 32, 41, 50,



© Jakob Greilhuber, Philipp Schepper, and Philip Wellnitz;
licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 41; pp. 41:1–41:20



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



52, 54, 59]. Moreover, the problems seem to come with a significant complexity but also sufficient structural properties to serve as a testing point for new techniques which frequently result in faster algorithms for the problems [20, 28, 61].

In 1993, Telle and Proskurowski introduced the general class of (σ, ρ) -DOMSET problems which capture several well-known vertex selection problems for appropriately chosen sets $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ [57, 58]. In this problem the input is an undirected graph and the task is to decide if we can select vertices such that (1), for every selected vertex, the number of selected neighbors is contained in the set σ and (2), for every unselected vertex, the number of selected neighbors is contained in the set ρ . Formally, for a graph G , decide if there exists a vertex set $S \subseteq V(G)$ such that, for all $v \in S$, we have $|N(v) \cap S| \in \sigma$, and, for all $v \notin S$, we have $|N(v) \cap S| \in \rho$. Such a set S is called a (σ, ρ) -set.

It is easy to see that (σ, ρ) -DOMSET captures classical Dominating Set when we set $\sigma = \mathbb{Z}_{\geq 0}$ and $\rho = \mathbb{Z}_{\geq 0} \setminus \{0\}$ and ask for a selection of bounded size. Moreover, with different requirements imposed on the size of the selection, we can also reformulate other problems such as Independent Set ($\sigma = \{0\}$ and $\rho = \mathbb{Z}_{\geq 0}$), Perfect Code ($\sigma = \{0\}$ and $\rho = \{1\}$), Induced q -regular Subgraph ($\sigma = \{q\}$ and $\rho = \mathbb{Z}_{\geq 0}$), Odd Domination ($\sigma = \{0, 2, \dots\}$ and $\rho = \{1, 3, \dots\}$), and many more. We refer to [8, 57] for a longer list of problems that can be described as (σ, ρ) -DOMSET.

Since (σ, ρ) -DOMSET generalizes many fundamental graph problems, the ultimate goal is to settle the complexity of (σ, ρ) -DOMSET for *all* (decidable) sets σ and ρ . We know that for many choices of σ and ρ the problem is NP-hard. Hence, we frequently either restrict the input to special graph classes or parameterize by some (structural) measure of the input (for example, the solution size).

One of the best explored structural parameters is treewidth [6, 18, 20, 26, 42, 45, 46, 51, 61] which measures how similar a graph is to a tree (see [19, Chapter 7] for a more thorough introduction). Many problems admit efficient algorithms on trees with a simple dynamic program. With treewidth as parameter, we can lift these programs to more general graphs and obtain fast algorithms especially compared to the running time obtained from Courcelle's Theorem [17]. For most of these problems the goal is to find the smallest constant c such that the respective problem can be solved in time $c^{\text{tw}} \cdot n^{\mathcal{O}(1)}$.

For problems parameterized by treewidth a perpetual improvement of the algorithm seems unlikely. After a few iterations, frequently conceptually new ideas seem to be necessary to obtain further improvements. Lokshantov, Marx, and Saurabh initiated a line of research that proves that such limitations are often not a shortcoming of the techniques at hand, but rather an inherent property of the problem itself [45]. For example, they prove that the known algorithm for Dominating Set [61] which takes time $3^{\text{tw}} \cdot n^{\mathcal{O}(1)}$ cannot be improved further unless the Strong Exponential-Time Hypothesis (SETH) [9, 40] fails.

Hence, the ultimate goal for (σ, ρ) -DOMSET is to show the following result (or to prove that no such constant exists in the respective setting):

For all sets σ and ρ , determine the constant $c_{\sigma, \rho}$ such that (σ, ρ) -DOMSET
can be solved in time $c_{\sigma, \rho}^{\text{tw}} \cdot n^{\mathcal{O}(1)}$
but not in time $(c_{\sigma, \rho} - \varepsilon)^{\text{tw}} \cdot n^{\mathcal{O}(1)}$ for any $\varepsilon > 0$, unless SETH fails.

For certain choices of the sets σ and ρ , some (partial) results of this form are already known [49]. A broad class of algorithms was given by van Rooij, Bodlaender, and Rossmanith [61] for the case of finite and cofinite sets. These algorithms were later improved by van Rooij [60]. Focke et al. [26, 24] recently introduced highly non-trivial techniques to improve these algorithms further for an infinite class of choices for σ and ρ . Moreover, Focke et al. additionally provide matching lower bounds for these new algorithms that rule out additional improvements [26, 25].

Beyond Finite and Cofinite Sets. Although the known results already capture large classes of problems, they are limited to finite and cofinite sets. This leaves open the entire range of infinite sets (with infinite complements), which contains not only “unstructured” sets like the set of prime numbers, for example, but also easy to describe and frequently used sets like the even or odd numbers, and arithmetic progressions in general.

One important example for families that are neither finite nor cofinite are *residue classes*. We say that a set $\tau \subseteq \mathbb{Z}_{\geq 0}$ is a residue class modulo m if there are two integers a and m such that $\tau = \{n \in \mathbb{Z}_{\geq 0} \mid n \equiv_m a\}$; we usually require that $0 \leq a < m$ for the canonical representation. Again, the two most natural residue classes are the even and odd numbers.

Surprisingly, for such infinite sets the complexity of (σ, ρ) -DOMSET is significantly underexplored and heavily fragmented even in the classical, non-parameterized, setting. This is especially surprising as this variant of the problem has direct applications in other fields like coding theory [13, 38].

Halldórsson, Kratochvíl, and Telle consider a variation of Independent Set where the unselected vertices have parity constraints [37]. They provide a complete dichotomy between polynomial-time solvable cases and NP-hard cases. Later the same group of authors considered the case where each of the sets comprises either the even or odd integers and proved similar hardness results [38]. Caro, Klostermeyer, and Goldwasser consider a variant of (σ, ρ) -DOMSET with residue classes as sets where they restrict the *closed* neighborhood of a vertex [11]. In this setting they prove new upper bounds for specific graphs classes including complements of powers of cycles and grid graphs. Fomin, Golovach, Kratochvíl, Kratsch, and Liedloff consider general graphs and provide *exponential-time* algorithms for general residue classes [27].

Although for the case of general sets some more results are known in the parameterized setting [1, 8, 30, 35], surprisingly few involve sets that are neither finite nor cofinite and the parameter treewidth. Gassner and Hatzl [33] consider the problem of residue domination where the sets are either all even or all odd numbers. They provide an algorithm for the problem with running time $2^{3\text{tw}} \cdot n^{\mathcal{O}(1)}$. Gassner and Hatzl also conjecture that their algorithm works when the sets have a larger modulus but unfortunately do not state the expected running time or the actual algorithm.

Chapelle in contrast provides a general algorithm for (σ, ρ) -DOMSET which covers *all* ultimately periodic sets¹ but, unfortunately, does not provide an explicit running time of the algorithm [14, 15].

In this work, we answer the main question from above and settle the complexity of (σ, ρ) -DOMSET for another large class of sets, namely for the residue classes.

► **Main Theorem 1.** *Write $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ for two residue classes modulo $m \geq 2$.*

Then, in time $m^{\text{tw}} \cdot |G|^{\mathcal{O}(1)}$ we can decide simultaneously for all s if the given graph G has a (σ, ρ) -set of size s when a tree decomposition of width tw is given with the input.

We remark that our algorithm does not only solve the decision version but also the maximization, minimization and exact version of (σ, ρ) -DOMSET.

Despite the fact that there are some pairs for which the decision version of (σ, ρ) -DOMSET is efficiently solvable (for example, the empty set is a trivial minimum solution if $0 \in \rho$), we prove that for all other “difficult” cases our algorithm is optimal even for the decision version and cannot be improved unless SETH fails. We refer to Definition 2.6 for a complete list of the “easy” pairs for which the decision version can be solved in polynomial time; to all other pairs we refer as “difficult”.

¹ A set τ is ultimately periodic if there is a finite automaton (over a unary alphabet) such that the length of the accepted words is precisely described by the set τ .

► **Main Theorem 2.** Write $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ for difficult residue classes modulo $m \geq 2$.

Unless *SETH* fails, for all $\varepsilon > 0$, there is no algorithm that can decide in time $(m - \varepsilon)^{\text{pw}} \cdot |G|^{\mathcal{O}(1)}$ whether the input graph G has a (σ, ρ) -set, when a path decomposition of width pw is given with the input.

Observe that our lower bound is for the larger parameter pathwidth, which immediately implies the result for the smaller parameter treewidth.

Our Contribution. Before we outline the formal ideas behind our results, we first highlight why these bounds are more surprising than they seem to be at the first glance.

To that end, let us take a deeper look at the algorithms of [24, 60]. Typically, the limiting factor for faster algorithms parameterized by treewidth is the number of states that have to be considered for each bag of the tree decomposition. For vertex selection problems, the state of a vertex is defined by two values: (1) whether it is selected and (2) how many selected neighbors it gets in some (partial) solution. To bound this latter number, we identify the largest “reasonable” state a vertex can have when it is selected and when it is unselected.

For finite sets σ and ρ this largest reasonable state is simply determined by the maximum of the respective sets, that is, we set $s_{\text{top}} = \max \sigma$ and $r_{\text{top}} = \max \rho$ as the largest reasonable number of neighbors, respectively.

Then, for a selected vertex, the allowed number of selected neighbors ranges from 0 to s_{top} , yielding $s_{\text{top}} + 1$ states for selected vertices. Similarly, we need to consider $r_{\text{top}} + 1$ states for unselected vertices. Combining the two cases, for each bag of the tree decomposition there are at most $(s_{\text{top}} + r_{\text{top}} + 2)^{\text{tw}+1}$ states to consider. Surprisingly, Focke et al. proved that, for an infinite number of finite sets, even at most $(t_{\text{top}} + 1)^{\text{tw}+1}$ of said states suffice, where $t_{\text{top}} = \max(s_{\text{top}}, r_{\text{top}})$ [26].² When $s_{\text{top}} = r_{\text{top}}$ this improves the algorithm by a factor of $2^{\text{tw}+1}$.

Similarly, for the case of residue classes with modulus m , the number of selected neighbors effectively ranges from 0 to $m - 1$ as for all larger values the behavior is equivalent to some smaller value. This gives us m states if a vertex is unselected and m states if a vertex is selected. Hence, the straight-forward bound for the number of states is $(2m)^{\text{tw}+1}$ which is a factor of $2^{\text{tw}+1}$ worse than the bound from the running time of our algorithm.

We remark that the most naive approach, for which we remove all integers from the sets σ and ρ that are larger than the number of vertices of the input graph and then apply the improved result by Focke et al. for finite sets, fails miserably. This approach would merely give an XP-algorithm as the size of the sets now depends on the size of the input graph.

Hence, it is far from trivial to obtain the claimed running time since the classical approach would not give something better than $(2m)^{\text{tw}} \cdot n^{\mathcal{O}(1)}$.

Our Techniques. Although we use the algorithmic result by Focke et al. as a basis for our upper bounds, our algorithm does not follow as an immediate corollary. There are two main challenges that we need to overcome to obtain the fast running time.

First, all previous results with *tight* bounds considered finite or cofinite sets. In this setting all integers (starting from some threshold) are somewhat equivalent in the sense that they are either all contained in the set or all not contained in the set. This makes defining a largest reasonable state quite convenient. For the residue classes this is not as easily possible as the integers change between membership and non-membership. Hence, we need an even more careful construction and analysis when improving upon the naive bound for the number of states.

² To keep notation simple, we omit the special case where the bound is $(t_{\text{top}} + 2)^{\text{tw}+1}$.

Second, it does not suffice to bound the number of states which the dynamic program considers, but we also need to be able to combine these states efficiently at the join nodes. Although this is a general issue when parameterizing by treewidth, until today there does not seem to be one solution which works in a black-box manner in all settings. Instead, we need to carefully design a new approach that takes care of the particular setting we consider which is based on but differs from the existing results for finite and cofinite sets.

For the lower bound result we are at a similar situation as for the upper bound; the setting is similar to what is known but still different. There are several known lower bounds for Dominating Set and its related problems, but these reductions are usually quite tailored to the specific problem and lack a modular construction – it is difficult to reuse existing results. The proofs are usually based on a direct reduction from k -SAT which introduces an additional overhead that obfuscates the high-level idea by technicalities (SAT has a running time bound of the form 2^n but we want a bound of the form c^{pw}).

We avoid this overhead by reducing from an appropriate Constraint Satisfaction Problem introduced by Lampis for precisely such settings [44]. Our results can be seen as one of the first applications of this new approach (outside the original setting) that can potentially also serve as a blue-print to simplify many other reductions and lower bound proofs or to directly obtain simpler results from scratch.

The Special Case of Parity Domination. When taking a closer look at the precise statement of our lower bound (and Definition 2.6), we are reminded that both results do not apply for the same set of pairs. Especially for the case of residue classes with modulus $m = 2$, our algorithm solves the minimization version, but Main Theorem 2 provides no matching lower bound. As we may solve the decision problem for these cases in polynomial-time via Gaussian elimination (see, for example, [2, 21, 34, 38, 56]), our lower bound explicitly excludes these cases by referring to them as “easy”.

Surprisingly, exactly these easy cases can be related to a single-player game called *Lights Out* that was published 1995. In this game the unassuming player is presented with a 5×5 grid of switches and lamps, some or all of them initially turned on, and the task is to turn off all lamps by pressing the switches. The catch is that every switch flips not only the state of its corresponding lamp (from “on” to “off” or vice-versa), but also the states of the neighboring lamps in the grid [5, 23].³

Since the order in which the buttons are pressed does not matter and every button has to be pressed at most once as a second press would undo the first operation, we can describe a *solution* to an initial configuration as a set of switches that need to be flipped to turn all lights off.

When we assume that initially all lights are turned on, then we can directly treat *Lights Out* as a variant of (σ, ρ) -DOMSET with $\sigma = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 0\}$ and $\rho = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 1\}$. We also refer to this problem, where the input is an arbitrary graph, as REFLEXIVE-ALLOFF since we assume that each switch triggers the corresponding lamp. When this is not the case but still all lights are initially turned on, we have $\sigma = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 1\} = \rho$ and refer to the problem as ALLOFF as the corresponding switch does not trigger the associated lamp.

Despite the fact that it is easy to find *some* solution for these two problems if one exists, the minimization versions do not have such a trivial answer and are known to be NP-complete [11, 38, 55]. Hence, we investigate the minimization versions for these two problems and complement the algorithmic result from Main Theorem 1 as follows.

³ Similar games have also been released under the names *Merlin* and *Orbit* [23].

► **Main Theorem 3.** *Unless SETH fails, for all $\varepsilon > 0$, there is no algorithm for each of the problems REFLEXIVE-ALLOFF and ALLOFF that can decide in time $(2 - \varepsilon)^{pw} \cdot |G|^{\mathcal{O}(1)}$ whether there exists a solution of arbitrary size (the size is given as input) for a graph G that is given with a path decomposition of width pw .*

Together with the lower bound for the general case, we conclude that our algorithm is the best possible, unless a major breakthrough for solving SAT happens.

Further Directions. When taking a step back, the results of this work serve two purposes which can be seen as starting points for further investigations and improvements.

- First, we settle the complexity of (σ, ρ) -DOMSET conclusively for the case of residue classes by providing matching upper and lower bounds.

We later list some candidates that might allow similar improvements. Such improvements would, similar to our results, extend the list of problems started by Focke et al. [26] where significant improvements for the supposedly optimal algorithms are possible.

- Second, in comparison to the fairly complicated results for the case of finite sets in [26], this work can be seen as a significantly simpler introduction to those techniques that are relevant to obtain faster algorithms by exploiting the structural properties of the sets. We believe that for many other (parameterized) problems – including but not limited to (σ, ρ) -DOMSET – the algorithms can be improved exponentially by using these new techniques.

In the following we list several possible directions that could serve as a next step on the route to a complete picture of the complexity of (σ, ρ) -DOMSET.

- A first natural case could be pairs of two residue classes with *different* moduli m_σ and m_ρ . Then, the natural structural parameter m (which is the modulus in our case) is the greatest common divisor of m_σ and m_ρ . In this setting the case $m = 1$ is also relevant as this does not directly imply that the sets contain all natural numbers.
- A different direction considers the combination of a residue class with a finite or cofinite set. Focke et al. show that representative sets [29, 43, 48, 53] can be used to speed up the algorithm even further for the case of cofinite sets [24]. Independently of finding the optimal algorithm to handle the join operation for representative sets, it is not even clear what the optimal running time should be in such a case.
- Caro and Jacobson [10] introduced the problem NON- $z(\bmod k)$ DOMINATING SET which can also be described as a (σ, ρ) -DOMSET problem where the sets are complements of residue classes, which is equivalent to a finite union of residue classes. For example, for $z = 0$ and $k = 3$, we set $\sigma = \{0, 1, 3, 4, \dots\} = \{0, 3, 6, \dots\} \cup \{1, 4, 7, \dots\}$ and $\rho = \{1, 2, 4, 5, 7, 8, \dots\} = \{1, 4, 7, \dots\} \cup \{2, 5, 8, \dots\}$. What is the optimal running time in this case?
- The general algorithm by Chapelle for the case when both sets are ultimately periodic has a running time single-exponential in treewidth despite being stated implicitly only [14, 15]. What is the best running time for an algorithm solving *all* cases of (σ, ρ) -DOMSET that are currently known to be fixed-parameter tractable?
- Are there more classes of sets for which there is an fpt algorithm parameterized by treewidth? Chapelle showed that once there are large gaps in the set, the problem becomes significantly harder [14, 15].

► **Theorem 1.1** ([14, Theorem 1] and [15, Théorème 3.3.1]). *Write σ for a set with arbitrarily large gaps between two consecutive elements (such that a gap of length t is at distance $\text{poly}(t)$ in σ), and write ρ for a cofinite set with $\min \sigma \geq 1$ and $\min \rho \geq 2$. Then, the problem (σ, ρ) -DOMSET is W[1]-hard when parameterized by the treewidth of the input graph.*

Examples are the two natural sets where $\sigma = \{2^i \mid i \in \mathbb{Z}_{\geq 0}\}$ or when σ is the set of all Fibonacci numbers [14]. We observe that this is one of the rare cases where a problem is W[1]-hard even when parameterizing by treewidth.

The classification by Chapelle is not a dichotomy result in the sense that it provides a full classification between the fpt cases and the ones that are W[1]-hard. For instance, what is the complexity for sets like $\sigma = \mathbb{Z}_{\geq 0} \setminus \{2^i \mid i \in \mathbb{Z}_{\geq 0}\}$ which have gaps of constant size only but are not ultimately periodic?

- With our results, there are improved algorithms for the case when the sets are finite, cofinite or residue classes. Nevertheless, the description of the exact running time is highly non-uniform, that is, the exact complexity explicitly depends on the underlying set. Can we describe the complexity of optimal algorithms in a compact form as, for example, done by Chapelle for the general algorithm via finite automata [14, 15]? This notation suffices to describe the state of a single vertex, but the representation of the structural insights leading to fewer states and faster algorithms remains open.
- Lastly, for which other problems besides (σ, ρ) -DOMSET can the techniques from our upper bounds (sparse languages and compression of vectors) be used to obtain faster algorithms? As the high-level idea of our lower bounds is quite modular, it should also be possible to use these concepts as blue-prints to achieve matching bounds for other problems as well.

2 Technical Overview

In this section we give a high-level overview of the results in this paper and outline the main technical contributions we use. We start by rigorously defining the main problem considered in this work and the property of a set being m -structured.

► **Definition 2.1** ((σ, ρ) -sets, (σ, ρ) -DOMSET). *Fix two non-empty sets σ and ρ of non-negative integers.*

For a graph G , a set $S \subseteq V(G)$ is a (σ, ρ) -set for G if and only if (1) for all $v \in S$, we have $|N(v) \cap S| \in \sigma$, and (2), for all $v \in V(G) \setminus S$, we have $|N(v) \cap S| \in \rho$.

The problem (σ, ρ) -DOMSET asks for a given graph G , whether there is a (σ, ρ) -set S or not.

We also refer to the problem above as the *decision* version. The problem naturally also admits related problems such as asking for a solution of a specific size, or for the smallest or largest solution, that is, the *minimization* and *maximization* version.

For the case of finite and cofinite sets, Focke et al. [24, 25] realized that the complexity of (σ, ρ) -DOMSET significantly changes (and allows faster algorithms) when σ and ρ exhibit a specific structure, which they refer to as *m-structured*.

► **Definition 2.2** (m -structured sets [24, Definition 3.2]). *Fix an integer $m \geq 1$. A set $\tau \subseteq \mathbb{Z}_{\geq 0}$ is m -structured if all numbers in τ are in the same residue class modulo m , that is, if there is an integer c^* such that $c \equiv_m c^*$ for all $c \in \tau$.*

Observe that every set τ is m -structured for $m = 1$. Therefore, one is usually interested in the largest m such that a set is m -structured. When considering two sets σ and ρ , we say that this pair is m -structured if each of the two sets is m -structured. More formally, assume that σ is m_σ -structured and ρ is m_ρ -structured. In this case the pair (σ, ρ) is m -structured where m is the greatest common divisor of m_σ and m_ρ . As in our case the sets σ and ρ are residue classes modulo $m \geq 2$, the sets are always m -structured.

In the following we first present the algorithmic result, which outlines the proof of Main Theorem 1. Afterward we move to the lower bounds where we consider Main Theorem 2 and finally we focus on the special case of Lights Out from Main Theorem 3.

2.1 Upper Bounds

The basic idea to prove the upper bound is to provide a dynamic programming algorithm that operates on a tree decomposition of the given graph. For each node of this decomposition we store all valid states, where each such state describes how a possible solution, i.e., a set of selected vertices, interacts with the bags of the corresponding node. We formalize this by the notion of a partial solution.

For a node t with associated bag X_t , we denote by V_t the set of vertices introduced in the subtree rooted at t and by G_t the graph induced by these vertices. We say that a set $S \subseteq V_t$ is a partial solution (for G_t) if

- for each vertex $v \in S \setminus X_t$, we have $|N(v) \cap S| \in \sigma$, and
- for each vertex $v \in V_t \setminus (S \cup X_t)$, we have $|N(v) \cap S| \in \rho$.

The solution is partial in the sense that there are no constraints imposed on the number of neighbors of the vertices in X_t , that is, only the vertices in $V_t \setminus X_t$ must have a valid number of neighbors.

We characterize the partial solutions by the states of the vertices in the bag. When σ and ρ are finite or cofinite sets, the largest reasonable state is included in the respective set, which is not necessarily the case for residue classes. Consider two fixed, residue classes σ and ρ modulo $m \geq 2$. Every selected vertex can have up to m different states and similarly, every unselected vertex can have m different states. Hence, for each bag, the number of relevant different partial solutions is bounded by $(2m)^{|X_t|}$.

High-level Idea. The crucial step to fast and efficient algorithms is to provide a better bound on the number of states for each bag when the sets σ and ρ are residue classes modulo $m \geq 2$. We denote by \mathbb{A} the set of all possible states a vertex might have in a valid solution. Then, let $L \subseteq \mathbb{A}^{X_t}$ be the set of all possible state-vectors corresponding to partial solutions for G_t . Our first goal is to show that $|L| \leq m^{|X_t|}$, which means that not all theoretically possible combinations of states can actually have a corresponding partial solution in the graph.

Moreover, we also need to be able to combine two partial solutions at the join nodes of the tree decomposition. For a fast join operation, it does not suffice to bound the size of L . This follows from the observation that the convolution algorithm used to handle the join operation does not depend on L but on the *space* where the states come from. In our case, the size of the space where L comes from is still $(2m)^{|X_t|}$, which is too large. To decrease this size, we observe that a significant amount of information about the states of the vertices can be inferred from other positions, that is, we can *compress* the vectors.

As a last step it remains to combine the states significantly faster than a naive algorithm. To efficiently compute the join, we use an approach based on the fast convolution techniques by van Rooij which was already used for the finite case [60]. However, we have to ensure that the compression of the vectors is actually compatible with the join operation, that

is, while designing the compression we already have to take in mind that we later join two (compressed) partial solutions together. Since the compressed vectors are significantly simpler, these states can now be combined much faster.

Bounding the Size of a Single Language. Recall that every partial solution S can be described by a state-vector $x \in \mathbb{A}^n$ where we abuse notation and set $n := |X_t|$. When x describes the partial solution S , we also say that S is a witness for x . We denote the set of the state-vectors of all partial solutions for G_t as L . We later refer to the set L as the realized language of (G_t, X_t) . To provide the improved bound on the size of L , we decompose each state-vector x into two vectors: The *selection-vector* of x , also called the σ -vector and denoted by $\vec{\sigma}(x)$, indicates whether each vertex in X_t is selected or not. The *weight-vector* of x , denoted by $\vec{w}(x)$, contains the number of selected neighbors of the vertices in X_t .

The key insight into the improved bound is that for two partial solutions of similar size (with regard to modulo m), the σ -vectors and the weight-vectors of these two solutions are orthogonal. This observation was already used to prove the improved bound when σ and ρ are finite [24]. We extend this result to the case of residue classes.

To formally state the key insight, we define the notion of a graph with portals.

► **Definition 2.3** (Graph with Portals; compare [24, Section 3]). *A graph with portals G is a pair (G', U) , where G' is a graph and $U \subseteq V(G')$. If $U = \{u_1, \dots, u_k\}$, then we also write (G', u_1, \dots, u_k) instead of (G', U) .*

If it is clear from the context, we also refer to a graph with portals simply as a graph.

Intuitively, one can think of G' being the graph G_t , and U the set X_t for a node t of a tree decomposition.

► **Lemma 2.4** (Compare [24, Lemma 4.3]). *Let σ and ρ denote two residue classes modulo $m \geq 2$. Let (G, U) be a graph with portals and let $L := L(G, U) \subseteq \mathbb{A}^U$ denote its realized language. Consider two strings $x, y \in L$ with witnesses $S_x, S_y \subseteq V(G)$ such that $|S_x \setminus U| \equiv_m |S_y \setminus U|$. Then, $\vec{\sigma}(x) \cdot \vec{w}(y) \equiv_m \vec{\sigma}(y) \cdot \vec{w}(x)$.*

To prove this result, we count edges between the vertices in S_x and the vertices in S_y in two different ways. We first count the edges based on their endpoint in S_x . These vertices can be partitioned into three groups: (1) the vertices contained in U , (2) the vertices outside U which are not in S_y , and (3) the vertices outside U which are in S_y . Then, the number of edges $|E(S_x \rightarrow S_y)|$ from S_x to S_y satisfies

$$|E(S_x \rightarrow S_y)| \equiv_m \min \rho \cdot |S_x \setminus (S_y \cup U)| + \min \sigma \cdot |(S_x \cap S_y) \setminus U| + \vec{\sigma}(x) \cdot \vec{w}(y)$$

because the sets σ and ρ are residue classes modulo m . When counting the edges based on their endpoint in S_y , the positions of x and y flip and the result follows. As this property enables us to prove that the size of L is small, we refer to this property as *sparse*.

Even though intuitively this orthogonality provides a reason why the size of the language is not too large, this does not result in a formal proof. However, when fixing which vertices are selected, that is, when fixing a σ -vector \vec{s} , then there is an even stronger restriction on the values of the weight-vectors. Instead of restricting the entire vector, it actually suffices to fix the vector on a certain number of positions which are described by some set S to which we refer as σ -defining set. If two σ -vectors of strings of the language agree on these positions from S , then *all* remaining positions of the two σ -vectors must be identical as well.

With the sparseness property we then show that it suffices to fix the σ -vectors on the positions from S (which then determines the values on \bar{S}), and the weight-vector on the positions from \bar{S} (which then determines the weight-vector on the positions from S). Formally, we prove Lemma 2.5 which mirrors [24, Lemma 4.9] in the case of residue classes.

► **Lemma 2.5** (Compare [24, Lemma 4.9]). *Let σ and ρ denote two residue classes modulo $m \geq 2$. Let $L \subseteq \mathbb{A}^n$ be a sparse language with a σ -defining set S for $\{\vec{\sigma}(x) \mid x \in L\}$. Then, for any two strings $x, y \in L$ with $\vec{\sigma}(x) = \vec{\sigma}(y)$, the positions \bar{S} uniquely characterize the weight vectors of x and y , that is, we have*

$$\vec{w}(x)[\bar{S}] = \vec{w}(y)[\bar{S}] \quad \text{implies} \quad \vec{w}(x) = \vec{w}(y).$$

With this result it is straight-forward to bound the size of a sparse language of dimension n . Our goal is to bound the number of weight-vectors that can be combined with a fixed σ -vector to form a valid type. Assume we fixed a σ -vector \vec{s} on the positions from S . Since this already determines the remaining positions of the σ -vector (even if we do not know the values a priori), the number of possible σ -vectors is at most $2^{|\bar{S}|}$. For the weight-vector there are m choices for each of the positions from \bar{S} . Then, the values for the positions from S are uniquely determined by those on \bar{S} because of the previous result. Using $m \geq 2$ this allows us to bound the size of a sparse language by

$$m^{|\bar{S}|} \cdot 2^{|\bar{S}|} \leq m^n.$$

Compressing Weight-Vectors. Based on the previous observations and results, we focus on the analysis for a fixed σ -vector \vec{s} . Though we could iterate over all at most $2^{|\bar{S}|}$ possible σ -vectors without dominating the running time, the final algorithm only considers the σ -vectors resulting from the underlying set L . Hence, we assume that all vectors in L share the same σ -vector \vec{s} .

When looking again at the bound for the size of L , it already becomes apparent how we can compress the weight-vectors. Recall that once we have fixed the entries of a weight-vector of some vector $x \in L$ at the positions of \bar{S} , the entries of the weight-vector on S are predetermined by Lemma 2.5. Hence, for the compressed vector we simply omit the entries on the positions in S , that is, the compressed weight-vector is the projection of the original weight-vector to the dimensions from \bar{S} .

It remains to recover the original vectors from their compression. As the implication from Lemma 2.5 actually does not provide the values for the positions on S , it seems tempting to store a single representative, which we refer to as *origin*-vector o to recover the omitted values for all compressed vectors. Unfortunately, this is not (yet) sufficient.

Observe that Lemma 2.5, which serves as basis for the compression, requires that the weight-vector u and the origin-vector o agree on the coordinates from \bar{S} . Therefore, it would be necessary to store one origin-vector for each possible choice of values on \bar{S} , which would not yield any improvement in the end.

In order to recover the values of the compressed weight-vector, we use our structural property from Lemma 2.4 once more. Intuitively, we use that changing the weight-vector at one position (from \bar{S} , in our case), has an effect on the value at some other position (from S , in our case). Based on this idea we define an auxiliary vector, which we refer to as *remainder*-vector. Intuitively, the entries of this vector capture the difference of the weight-vector u and the origin-vector o on the positions in \bar{S} . By the previous observation this also encodes how much these two vectors u and o differ on the positions from S . This remainder-vector then allows us to efficiently decompress the compressed weight-vectors again. In consequence, the final compression reduces the size of the space where the weight-vectors are chosen from, which is a prerequisite for the last part of the algorithm.

Faster Join Operations. To obtain the fast join operation, we apply the known convolution techniques by van Rooij [60]. As the convolution requires that all operations are done modulo some small number, we can directly apply it as every coordinate of the compressed vector is computed modulo m . As the convolution operates in the time of the space where the vectors are from, we obtain an overall running time of $m^{|X_\varepsilon|}$ for the join operation.

The final algorithm is then a dynamic program where the procedures for all nodes except the join node follow the standard procedure. For the join node, we iterate over all potential σ -vectors of the combined language, then join the compressed weight-vectors, and finally output the union of their decompositions.

By designing the algorithm such that we consider solutions of a certain size, we achieve that the considered languages are sparse and thus, the established machinery provides the optimal bound for the running time. In total, we obtain Main Theorem 1.

► **Main Theorem 1.** *Write $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ for two residue classes modulo $m \geq 2$.*

Then, in time $m^{\text{tw}} \cdot |G|^{\mathcal{O}(1)}$ we can decide simultaneously for all s if the given graph G has a (σ, ρ) -set of size s when a tree decomposition of width tw is given with the input.

2.2 Lower Bounds

After establishing the upper bounds, we focus on proving matching lower bounds, that is, we prove the previous algorithm to be optimal under SETH. For all difficult cases, we provide a general lower bound and for the easy cases that are solvable in polynomial time but are non-trivial, we prove a lower bound for the minimization version by a separate reduction. In the following we first focus on the difficult cases.

► **Definition 2.6 (Easy and Difficult Cases).** *Let σ and ρ be two residue classes. We say that this pair is easy if $0 \in \rho$ or*

- $\sigma = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 0\}$ and $\rho = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 1\}$, or
- $\sigma = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 1\}$ and $\rho = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 1\}$.

Otherwise, we say that the pair is difficult.

Clearly the case $0 \in \rho$ is trivial since the empty set is a valid solution. For the other two cases we can formulate the problem as a system of linear equations over \mathbb{F}_2 : for each vertex we create a variable indicating the selection status and introduce one appropriately chosen constraint involving the neighboring vertices. Then, Gaussian elimination provides a solution in polynomial time. We refer to [2, 34, 38, 56] for a formal proof. Thus, unless mentioned otherwise, we henceforth focus on the difficult cases.

Starting from the first SETH-based lower bounds when parameterizing by treewidth by Lokshtanov, Marx and Saurabh [45] (see also references in [44] for other applications) many reductions suffered from the following obstacle: SETH provides a lower bound of the form $(2 - \varepsilon)^n$ whereas for most problems a lower bound of the form $(c - \varepsilon)^{\text{tw}}$ is needed for some integer $c > 2$. To bridge this gap, several technicalities are needed to obtain the bound with the correct base. Lampis introduced the problem (family) q -CSP- B , which hides these technicalities and allows for cleaner reductions. This problem generalizes q -SAT such that every variable can now take B different values, that is, for $B = 2$ this is the classical q -SAT problem. Formally q -CSP- B is defined as follows.

► **Definition 2.7 (q -CSP- B [44]).** *Fix two numbers $q, B \geq 2$. An instance of q -CSP- B is a tuple (X, \mathcal{C}) that consists of a set X of n variables having the domain $D = [1..B]$ each, and a set \mathcal{C} of constraints on X . A constraint C is a pair $(\text{scp}(C), \text{acc}(C))$ where $\text{scp}(C) \in X^q$ is the scope of C and $\text{acc}(C) \subseteq D^q$ is the set of accepted states.*

41:12 Residue Domination in Bounded-Treewidth Graphs

The task of the problem is to decide if (X, \mathcal{C}) is satisfiable, that is, decide if there exists an assignment $\pi: X \rightarrow D$ such that, for all constraints C with $\text{scp}(C) = (v_{\lambda_1}, \dots, v_{\lambda_q})$ it holds that $(\pi(v_{\lambda_1}), \dots, \pi(v_{\lambda_q})) \in \text{acc}(C)$.

In other words, the constraints specify valid assignments for the variables, and we are looking for a variable assignment that satisfies all constraints.

Apart from introducing this problem, Lampis also proved a conditional lower bound based on SETH which allows us to base our reduction on this special type of CSP.

► **Theorem 2.8** ([44, Theorem 3.1]). *For any $B \geq 2, \varepsilon > 0$ we have the following: assuming SETH, there is a q such that n -variable q -CSP- B with ℓ constraints cannot be solved in time $(B - \varepsilon)^n \cdot (n + \ell)^{\mathcal{O}(1)}$.*

To obtain the correct lower bound the most suitable version of q -CSP- B can be used, which then hides the unwanted technicalities.

In our case we cover numerous (actually infinitely many) problems. This creates many positions in the potential proof where (unwanted) properties of the sets σ and ρ have to be circumvented or exploited. In order to minimize these places and to make use of the special starting problem, we split the proof in two parts. This concept of splitting the reduction has already proven to be successful for several other problems [18, 24, 47, 48].

As synchronizing point, we generalize the known (σ, ρ) -DOMSET problem where we additionally allow that *relations* are added to the graph. Therefore, we refer to this problem as (σ, ρ) -DOMSET^{REL}. Intuitively one can think of these relations as constraints that observe a predefined set of vertices, which we refer to as *scope*, and enforce that only certain ways of selecting these vertices are allowed in a valid solution. To formally state this intermediate problem, we first define the notion of a *graph with relations*.

► **Definition 2.9** (Graph with Relations [25, Definition 4.1]). *We define a graph with relations as a tuple $G = (V, E, \mathcal{C})$, where V is a set of vertices, E is a set of edges on V , and \mathcal{C} is a set of relational constraints, that is, each $C \in \mathcal{C}$ is in itself a tuple $(\text{scp}(C), \text{acc}(C))$. Here the scope $\text{scp}(C)$ of C is an unordered tuple of $|\text{scp}(C)|$ vertices from V . Then, $\text{acc}(C) \subseteq 2^{\text{scp}(C)}$ is a $|\text{scp}(C)|$ -ary relation specifying possible selections within $\text{scp}(C)$. We also say that C observes $\text{scp}(C)$.*

The size of G is $|G| := |V| + \sum_{C \in \mathcal{C}} |\text{acc}(C)|$. Slightly abusing notation, we usually do not distinguish between G and its underlying graph (V, E) . We use G to refer to both objects depending on the context.

We define the treewidth and pathwidth of a graph with relations as the corresponding measure of the modified graph that is obtained by replacing all relations by a clique on the vertices from the scope.

We lift the notion of (σ, ρ) -set from Definition 2.1 to graphs with relations by requiring that every relation has to be satisfied as well. Hence, the definition of (σ, ρ) -DOMSET^{REL} follows naturally. These definitions are a reformulation of [25, Definition 4.3 and 4.8].

► **Definition 2.10** ((σ, ρ) -Sets of a Graph with Relations, (σ, ρ) -DOMSET^{REL}). *Fix two non-empty sets σ and ρ of non-negative integers.*

For a graph with relations $G = (V, E, \mathcal{C})$, a set $S \subseteq V$ is a (σ, ρ) -set of G if and only if (1) S is a (σ, ρ) -set of the underlying graph (V, E) and (2) for every $C \in \mathcal{C}$, the set S satisfies $S \cap \text{scp}(C) \in \text{acc}(C)$. We use $|G|$ as the size of the graph and say that the arity of G is the maximum arity of a relation of G .

The problem (σ, ρ) -DOMSET^{REL} asks for a given graph with relations $G = (V, C, \mathcal{C})$, whether there is such a (σ, ρ) -set or not.

With this intermediate problem, we can now formally state the two parts of our lower bound proof. The first step embeds the q -CSP- B problem (for appropriately chosen B) into the graph problem (σ, ρ) -DOMSET^{REL}. We design the reduction in such a way that the resulting instance has a small pathwidth (namely, roughly equal to the number of variables). Combined with the conditional lower bound for q -CSP- B based on SETH from Theorem 2.8, we prove the following intermediate lower bound.

► **Lemma 2.11.** *Let σ and ρ be two residue classes modulo $m \geq 2$.*

Then, for all $\varepsilon > 0$, there is a constant d such that (σ, ρ) -DOMSET^{REL} on instances of arity at most d cannot be solved in time $(m - \varepsilon)^{\text{pw}} \cdot |G|^{\mathcal{O}(1)}$, where pw is the width of a path decomposition provided with the input instance G , unless SETH fails.

For the second step, we then remove the relations from the constructed (σ, ρ) -DOMSET^{REL} instance, to obtain a reduction to the (σ, ρ) -DOMSET problem. Observe that the construction from Lemma 2.11 works for the general case (even when $0 \in \rho$ is allowed). Hence, our second step now exploits that the sets are difficult.

► **Lemma 2.12.** *Let σ and ρ be two difficult residue classes modulo m . For all constants d , there is a polynomial-time reduction from (σ, ρ) -DOMSET^{REL} on instances with arity d given with a path decomposition of width pw to (σ, ρ) -DOMSET on instances given with a path decomposition of width $\text{pw} + \mathcal{O}(2^d)$.*

Combining these two intermediate results directly leads to the proof of Main Theorem 2.

► **Main Theorem 2.** *Write $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ for difficult residue classes modulo $m \geq 2$.*

Unless SETH fails, for all $\varepsilon > 0$, there is no algorithm that can decide in time $(m - \varepsilon)^{\text{pw}} \cdot |G|^{\mathcal{O}(1)}$ whether the input graph G has a (σ, ρ) -set, when a path decomposition of width pw is given with the input.

Proof. Assume we are given a faster algorithm for (σ, ρ) -DOMSET for some $\varepsilon > 0$. Let d be the constant from Lemma 2.11 such that there is no algorithm solving (σ, ρ) -DOMSET^{REL} in time $(m - \varepsilon)^{\text{pw}} \cdot |G|^{\mathcal{O}(1)}$ when the input instance G is given with a path decomposition of width pw .

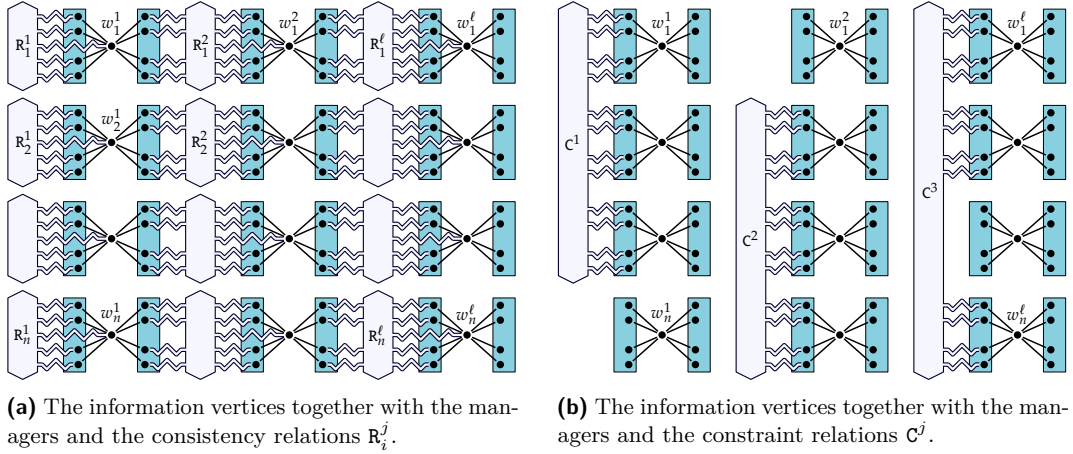
Consider an instance G of (σ, ρ) -DOMSET^{REL} with arity d along with a path decomposition of width $\text{pw}(G)$. We use Lemma 2.12 to transform this instance into an instance G' of (σ, ρ) -DOMSET with a path decomposition of width $\text{pw}(G') = \text{pw}(G) + \mathcal{O}(2^d)$.

We apply the fast algorithm for (σ, ρ) -DOMSET to the instance G' which correctly outputs the answer for the original instance G of (σ, ρ) -DOMSET^{REL}. The running time of this entire procedure is

$$|G|^{\mathcal{O}(1)} + (m - \varepsilon)^{\text{pw}(G')} \cdot |G'|^{\mathcal{O}(1)} = (m - \varepsilon)^{\text{pw}(G) + \mathcal{O}(2^d)} \cdot |G|^{\mathcal{O}(1)} = (m - \varepsilon)^{\text{pw}(G)} \cdot |G|^{\mathcal{O}(1)}$$

since d is a constant only depending on ε . This contradicts SETH and concludes the proof. ◀

The following highlights the main technical contributions leading to the results from Lemmas 2.11 and 2.12.



■ **Figure 1** A depiction of the construction from the lower bound where $m = 5$, $n = 4$, and $\ell = 3$.

Step 1: Encoding the CSP as a Graph Problem

Focke et al. already established the corresponding intermediate result when σ and ρ are finite [25]. Hence, we could try to reuse their lower bound for $(\hat{\sigma}, \hat{\rho})\text{-DOMSET}^{\text{REL}}$ for two finite sets $\hat{\sigma} \subseteq \sigma$ and $\hat{\rho} \subseteq \rho$. However, since σ and ρ are residue classes, several solutions could be indistinguishable from each other (not globally but locally from the perspective of a single vertex) which would result in unpredictable behavior of the construction. Thus, we need to come up with a new intermediate lower bound.

To prove this lower bound for $(\sigma, \rho)\text{-DOMSET}^{\text{REL}}$, we provide a reduction from $q\text{-CSP-}B$ where $B = m$ but reuse some ideas from the known lower bounds in [18, 25, 47, 48]. This allows for a much cleaner reduction (especially compared to the one from [25]) that focuses on the conversion of a constraint satisfaction problem into a vertex selection problem without having to deal with technicalities. Consult Figure 1 for an illustration of the high-level idea of the construction.

Consider a $q\text{-CSP-}m$ instance I with n variables and ℓ constraints. To achieve a low treewidth (or actually pathwidth), we construct a graph with $n \cdot \ell$ vertices, which we refer to as *information vertices*, that are arranged as an n times ℓ grid; rows corresponding to variables and columns corresponding to constraints. We refer to the information vertex from row i and column j as w_i^j . We encode the m different values of each variable by the states of the information vertices in the graph.

To provide sufficiently many neighbors to these information vertices, we introduce *managers*. In our case a manager consists of $2n$ blocks, n left blocks and n right blocks, and each block can provide up to $m - 1$ neighbors to a single vertex. We create one manager for each column (i.e., constraint) and assign one left block and one right block to each information vertex. Then, we make each information vertex adjacent to the two associated blocks by $m - 1$ edges each.

We use the number of selected neighbors from the left block to determine the state of an information vertex (though the vertex might have selected neighbors in the right block too). This directly relates the states of the information vertices to the variable assignments.

Recall that we create a separate manager for each column and that the managers are not connected to each other. Thus, despite the mentioned correspondence, even for a single row the information vertices can have different states. Phrased differently, the encoded

assignment is not necessarily consistent. To keep treewidth low, we cannot simply add a single big relation for each row enforcing the intended behavior. Instead, for each row i , we add a small *consistency relation* R_i^j between every two consecutive columns j and $j + 1$.

The relation R_i^j ensures the consistency between the information vertices w_i^j and w_i^{j+1} , and thus, additionally observes the right block of w_i^j and the left block of w_i^{j+1} . First, R_i^j ensures that information vertex w_i^j is unselected. Now assume that w_i^j has b_1 selected neighbors in its right block and w_i^{j+1} has b_2 selected neighbors in its left block. Then, relation R_i^j ensures that b_2 complements b_1 in the sense that $b_2 = \min \rho - b_1 \pmod m$, that is, b_2 is the smallest number such that $b_1 + b_2 \equiv_m \min \rho$.

It remains to analyze the influence of the information vertices themselves on the consistency of the encoded assignment. By our construction, information vertex w_i^j receives b_0 neighbors from its left block of the manager, receives b_1 neighbors from its right block of the manager, and receives no other neighbors. Since we consider a solution, vertex w_i^j must have a valid number of neighbors, that is, the solution must satisfy $b_0 + b_1 \in \rho$. Since ρ is a residue class modulo m , we get $b_0 + b_1 \equiv_m \min \rho$ which implies that $b_1 = \min \rho - b_0 \pmod m$. When combining this with the observation from the previous paragraph, where we consider two different information vertices, we get $b_2 = \min \rho - (\min \rho - b_0 \pmod m) \pmod m$ and hence, $b_2 = b_0 \pmod m$ which implies that all information vertices of one row have the same state.

As a last step we encode the constraints of the CSP instance. For each constraint C_j we add one *constraint relation* C^j which observes, for each variable appearing in C_j , the neighbors of the corresponding information vertex in the left block of the manager (they are needed to infer the state of the information vertices). The relation C^j then accepts a selection of these vertices if and only if it corresponds to a satisfying assignment.

This concludes the lower bound for the intermediate problem (σ, ρ) -DOMSET^{REL}. Next, we remove the relations and replace them by appropriate gadgets to lift the result to (σ, ρ) -DOMSET.

Step 2: Realizing the Relations

Formally, the second step is a reduction from (σ, ρ) -DOMSET^{REL} to (σ, ρ) -DOMSET. We replace each relation by a suitable gadget that precisely mimics the behavior of the original relation, that is, we *realize* the relation. The realization gadget accepts a selection of vertices if and only if the original relation also accepted this selection. Moreover, such a gadget must not add any selected neighbors to a vertex from the scope, as that could affect the existence of a solution (in the positive but also in the negative).

Curticapean and Marx [18] show that just two types of relations suffice to realize arbitrary relations. Focke et al. prove that for (σ, ρ) -DOMSET only $\text{HW}_{=1}$ relations are needed [25, Corollary 8.8], that is, once we can realize such $\text{HW}_{=1}$ relations, then every relation can be realized. The $\text{HW}_{=1}$ relation accepts if exactly one vertex from the scope of the relation is selected, that is, if the Hamming weight of the σ -vector is exactly one. We strengthen this result further by using an observation from [47] such that only realizations of $\text{HW}_{=1}$ with arity one, two or three are needed.

To realize these relations, we use an auxiliary relation. For some set τ , the relation $\text{HW}_{\in \tau}$ accepts, if and only if the number of selected vertices from the scope of the relation, i.e., the Hamming weight of the σ -vector, is contained in τ . Once we set $\tau - k := \{t - k \mid t \in \tau\}$ to simplify notation, our main results for realizing relations reads as follows.

► **Lemma 2.13.** *Let σ and ρ be two difficult residue classes modulo m . Then, the relation $\text{HW}_{\in \rho - \min \rho + 1}$ can be realized.*

When σ and ρ are difficult we have $m \geq 3$, and hence this gadget directly gives $\text{HW}_{=1}$ when restricting to arity at most 3 as $\min \rho + 1, \min \rho + 2 \notin \rho$. Thus, together with the previous intermediate lower bound this concludes the proof of the lower bound.

For $m = 2$ the decision version is easy, so we cannot expect to realize the $\text{HW}_{=1}$ relation. So, we consider the *minimization version* instead and focus on the two non-trivial cases; for $\rho = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 1\}$, we consider $\sigma = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 1\}$ and $\sigma = \{x \in \mathbb{Z}_{\geq 0} \mid x \equiv_2 0\}$. For the lower bounds from Main Theorem 3, we modify the known NP-hardness reductions by Sutner [55] to keep pathwidth low and to obtain the matching bounds.

References

- 1 Jochen Alber, Hans L. Bodlaender, Henning Fernau, Ton Kloks, and Rolf Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002. doi:10.1007/S00453-001-0116-5.
- 2 Marlow Anderson and Todd Feil. Turning lights out with linear algebra. *Mathematics Magazine*, 71(4):300–303, 1998. doi:10.1080/0025570X.1998.11996658.
- 3 Ferhat Ay, Manolis Kellis, and Tamer Kahveci. Submap: Aligning metabolic pathways with subnetwork mappings. *J. Comput. Biol.*, 18(3):219–235, 2011. PMID: 21385030. doi:10.1089/cmb.2010.0280.
- 4 Lukas Barth, Benjamin Niedermann, Martin Nöllenburg, and Darren Strash. Temporal map labeling: a new unified framework with experiments. In Siva Ravada, Mohammed Eunus Ali, Shawn D. Newsam, Matthias Renz, and Goce Trajcevski, editors, *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2016, Burlingame, California, USA, October 31 - November 3, 2016*, pages 23:1–23:10. ACM, 2016. doi:10.1145/2996913.2996957.
- 5 Abraham Berman, Franziska Borer, and Norbert Hungerbühler. Lights out on graphs. *Mathematische Semesterberichte*, 68(2):237–255, 2021. doi:10.1007/s00591-021-00297-5.
- 6 Hans L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 226–234. ACM, 1993. doi:10.1145/167088.167161.
- 7 Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63 of *LIPICs*, pages 8:1–8:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.IPEC.2016.8.
- 8 Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theor. Comput. Sci.*, 511:66–76, 2013. doi:10.1016/J.TCS.2013.01.009.
- 9 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In Jianer Chen and Fedor V. Fomin, editors, *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, volume 5917 of *Lecture Notes in Computer Science*, pages 75–85. Springer, 2009. doi:10.1007/978-3-642-11269-0_6.
- 10 Yair Caro and Michael S. Jacobson. On non- $z \pmod k$ dominating sets. *Discuss. Math. Graph Theory*, 23(1):189–199, 2003. doi:10.7151/dmgt.1195.
- 11 Yair Caro, William F. Klostermeyer, and John L. Goldwasser. Odd and residue domination numbers of a graph. *Discuss. Math. Graph Theory*, 21(1):119–136, 2001. doi:10.7151/dmgt.1137.
- 12 David Cattanéo and Simon Perdrix. Parameterized complexity of weak odd domination problems. In Leszek Gasieniec and Frank Wolter, editors, *Fundamentals of Computation Theory - 19th International Symposium, FCT 2013, Liverpool, UK, August 19-21, 2013. Proceedings*, volume 8070 of *Lecture Notes in Computer Science*, pages 107–120. Springer, 2013. doi:10.1007/978-3-642-40164-0_13.

- 13 David Cattanéo and Simon Perdrix. The parameterized complexity of domination-type problems and application to linear codes. In T. V. Gopal, Manindra Agrawal, Angsheng Li, and S. Barry Cooper, editors, *Theory and Applications of Models of Computation - 11th Annual Conference, TAMC 2014, Chennai, India, April 11-13, 2014. Proceedings*, volume 8402 of *Lecture Notes in Computer Science*, pages 86–103. Springer, 2014. doi:10.1007/978-3-319-06089-7_7.
- 14 Mathieu Chapelle. Parameterized complexity of generalized domination problems on bounded tree-width graphs. *CoRR*, abs/1004.2642v5, 2010. doi:10.48550/arxiv.1004.2642.
- 15 Mathieu Chapelle. *Décompositions de graphes : quelques limites et obstructions. (Graphs decompositions: some limits and obstructions)*. PhD thesis, University of Orléans, France, 2011. URL: <https://tel.archives-ouvertes.fr/tel-00659666>.
- 16 E. Cockayne and S. Hedetniemi. Optimal domination in graphs. *IEEE Transactions on Circuits and Systems*, 22(11):855–857, 1975. doi:10.1109/TCS.1975.1083994.
- 17 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.
- 18 Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1650–1669. SIAM, 2016. doi:10.1137/1.9781611974331.CH113.
- 19 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 20 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Trans. Algorithms*, 18(2):17:1–17:31, 2022. doi:10.1145/3506707.
- 21 Yevgeniy Dodis and Peter Winkler. Universal configurations in light-flipping games. In S. Rao Kosaraju, editor, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA*, pages 926–927. ACM/SIAM, 2001. URL: <http://dl.acm.org/citation.cfm?id=365411.365812>.
- 22 Devdatt P. Dubhashi, Alessandro Mei, Alessandro Panconesi, Jaikumar Radhakrishnan, and Aravind Srinivasan. Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA*, pages 717–724. ACM/SIAM, 2003. URL: <http://dl.acm.org/citation.cfm?id=644108.644226>.
- 23 Rudolf Fleischer and Jiajin Yu. A survey of the game “Lights Out!”. In Andrej Brodnik, Alejandro López-Ortiz, Venkatesh Raman, and Alfredo Viola, editors, *Space-Efficient Data Structures, Streams, and Algorithms - Papers in Honor of J. Ian Munro on the Occasion of His 66th Birthday*, volume 8066 of *Lecture Notes in Computer Science*, pages 176–198. Springer, 2013. doi:10.1007/978-3-642-40273-9_13.
- 24 Jacob Focke, Dániel Marx, Fionn Mc Inerney, Daniel Neuen, Govind S. Sankar, Philipp Schepper, and Philip Wellnitz. Tight complexity bounds for counting generalized dominating sets in bounded-treewidth graphs Part I: Algorithmic results. *CoRR*, abs/2211.04278v2, 2023. doi:10.48550/arXiv.2211.04278.
- 25 Jacob Focke, Dániel Marx, Fionn Mc Inerney, Daniel Neuen, Govind S. Sankar, Philipp Schepper, and Philip Wellnitz. Tight complexity bounds for counting generalized dominating sets in bounded-treewidth graphs Part II: Hardness results. *CoRR*, abs/2306.03640, 2023. doi:10.48550/arXiv.2306.03640.
- 26 Jacob Focke, Dániel Marx, Fionn Mc Inerney, Daniel Neuen, Govind S. Sankar, Philipp Schepper, and Philip Wellnitz. Tight complexity bounds for counting generalized dominating sets in bounded-treewidth graphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3664–3683. SIAM, 2023. doi:10.1137/1.9781611977554.ch140.

- 27 Fedor V. Fomin, Petr A. Golovach, Jan Kratochvíl, Dieter Kratsch, and Mathieu Liedloff. Sort and search: Exact algorithms for generalized domination. *Inf. Process. Lett.*, 109(14):795–798, 2009. doi:10.1016/J.IPL.2009.03.023.
- 28 Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. A measure & conquer approach for the analysis of exact algorithms. *J. ACM*, 56(5):25:1–25:32, 2009. doi:10.1145/1552285.1552286.
- 29 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.
- 30 Fedor V. Fomin and Dimitrios M. Thilikos. Dominating sets in planar graphs: Branch-width and exponential speed-up. *SIAM J. Comput.*, 36(2):281–309, 2006. doi:10.1137/S0097539702419649.
- 31 Andrei Gagarin and Pdraig Corcoran. Multiple domination models for placement of electric vehicle charging stations in road networks. *Comput. Oper. Res.*, 96:69–79, 2018. doi:10.1016/j.cor.2018.03.014.
- 32 Serge Gaspers, Dieter Kratsch, Mathieu Liedloff, and Ioan Todinca. Exponential time algorithms for the minimum dominating set problem on some graph classes. *ACM Trans. Algorithms*, 6(1):9:1–9:21, 2009. doi:10.1145/1644015.1644024.
- 33 Elisabeth Gassner and Johannes Hatzl. A parity domination problem in graphs with bounded treewidth and distance-hereditary graphs. *Computing*, 82(2-3):171–187, July 2008. doi:10.1007/s00607-008-0005-8.
- 34 John Goldwasser, William Klostermeyer, and George Trapp. Characterizing switch-setting problems. *Linear and Multilinear Algebra*, 43(1-3):121–135, 1997. doi:10.1080/03081089708818520.
- 35 Petr A. Golovach, Jan Kratochvíl, and Ondrej Suchý. Parameterized complexity of generalized domination problems. *Discret. Appl. Math.*, 160(6):780–792, 2012. doi:10.1016/J.DAM.2010.11.012.
- 36 Jakob Greilhuber. Shining light on periodic dominating sets in bounded-treewidth graphs. Master’s thesis, TU Wien, 2024. doi:10.34726/hss.2024.120579.
- 37 Magnús M. Halldórsson, Jan Kratochvíl, and Jan Arne Telle. Independent sets with domination constraints. *Discret. Appl. Math.*, 99(1-3):39–54, 2000. doi:10.1016/S0166-218X(99)00124-9.
- 38 Magnús M. Halldórsson, Jan Kratochvíl, and Jan Arne Telle. Mod-2 independence and domination in graphs. *Int. J. Found. Comput. Sci.*, 11(3):355–363, 2000. doi:10.1142/S0129054100000272.
- 39 Stephen T. Hedetniemi and Renu C. Laskar. Bibliography on domination in graphs and some basic definitions of domination parameters. *Discret. Math.*, 86(1-3):257–277, 1990. doi:10.1016/0012-365X(90)90365-0.
- 40 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.
- 41 Samir Khuller, Manish Purohit, and Kanthi K. Sarpatwar. Analyzing the optimal neighborhood: Algorithms for budgeted and partial connected dominating set problems. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1702–1713. SIAM, 2014. doi:10.1137/1.9781611973402.123.
- 42 Tuukka Korhonen. A single-exponential time 2-approximation algorithm for treewidth. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 184–192. IEEE, 2021. doi:10.1109/FOCS52979.2021.00026.
- 43 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi:10.1145/3390887.
- 44 Michael Lampis. Finer tight bounds for coloring on clique-width. *SIAM J. Discret. Math.*, 34(3):1538–1558, 2020. doi:10.1137/19M1280326.

- 45 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, April 2018. doi:10.1145/3170442.
- 46 Dániel Marx. Four shorts stories on surprising algorithmic uses of treewidth. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 129–144. Springer, 2020. doi:10.1007/978-3-030-42071-0_10.
- 47 Dániel Marx, Govind S. Sankar, and Philipp Schepper. Degrees and gaps: Tight complexity results of general factor problems parameterized by treewidth and cutwidth. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPICs*, pages 95:1–95:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ICALP.2021.95.
- 48 Dániel Marx, Govind S. Sankar, and Philipp Schepper. Anti-factor is FPT parameterized by treewidth and list size (but counting is hard). In Holger Dell and Jesper Nederlof, editors, *17th International Symposium on Parameterized and Exact Computation, IPEC 2022, September 7-9, 2022, Potsdam, Germany*, volume 249 of *LIPICs*, pages 22:1–22:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.IPEC.2022.22.
- 49 Mohsen Alambardar Meybodi, Fedor V. Fomin, Amer E. Mouawad, and Fahad Panolan. On the parameterized complexity of $[1, j]$ -domination problems. *Theor. Comput. Sci.*, 804:207–218, 2020. doi:10.1016/j.tcs.2019.11.032.
- 50 Neeldhara Misra and Piyush Rathi. The parameterized complexity of dominating set and friends revisited for structured graphs. In René van Bevern and Gregory Kucherov, editors, *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019, Novosibirsk, Russia, July 1-5, 2019, Proceedings*, volume 11532 of *Lecture Notes in Computer Science*, pages 299–310. Springer, 2019. doi:10.1007/978-3-030-19955-5_26.
- 51 Karolina Okrasa and Pawel Rzazewski. Fine-grained complexity of graph homomorphism problem for bounded-treewidth graphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1578–1590. SIAM, 2020. doi:10.1137/1.9781611975994.97.
- 52 Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Solving dominating set in larger classes of graphs: FPT algorithms and polynomial kernels. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 694–705. Springer, 2009. doi:10.1007/978-3-642-04128-0_62.
- 53 Hadas Shachnai and Meirav Zehavi. Representative families: A unified tradeoff-based approach. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 786–797. Springer, 2014. doi:10.1007/978-3-662-44777-2_65.
- 54 Shay Solomon and Amitai Uzrad. Dynamic $((1+\epsilon) \ln n)$ -approximation algorithms for minimum set cover and dominating set. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1187–1200. ACM, 2023. doi:10.1145/3564246.3585211.
- 55 Klaus Sutner. Additive automata on graphs. *Complex Syst.*, 2(6), 1988. URL: http://www.complex-systems.com/abstracts/v02_i06_a03.html.
- 56 Klaus Sutner. Linear cellular automata and the garden-of-eden. *The Mathematical Intelligencer*, 11(2):49–53, 1989. doi:10.1007/BF03023823.
- 57 Jan Arne Telle. Complexity of domination-type problems in graphs. *Nord. J. Comput.*, 1(1):157–171, 1994.

- 58 Jan Arne Telle and Andrzej Proskurowski. Practical algorithms on partial k-trees with an application to domination-like problems. In Frank K. H. A. Dehne, Jörg-Rüdiger Sack, Nicola Santoro, and Sue Whitesides, editors, *Algorithms and Data Structures, Third Workshop, WADS '93, Montréal, Canada, August 11-13, 1993, Proceedings*, volume 709 of *Lecture Notes in Computer Science*, pages 610–621. Springer, 1993. doi:10.1007/3-540-57155-8_284.
- 59 Kuo-Hui Tsai and Wen-Lian Hsu. Fast algorithms for the dominating set problem on permutation graphs. In Tetsuo Asano, Toshihide Ibaraki, Hiroshi Imai, and Takao Nishizeki, editors, *Algorithms, International Symposium SIGAL '90, Tokyo, Japan, August 16-18, 1990, Proceedings*, volume 450 of *Lecture Notes in Computer Science*, pages 109–117. Springer, 1990. doi:10.1007/3-540-52921-7_60.
- 60 Johan M. M. van Rooij. Fast algorithms for join operations on tree decompositions. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 262–297. Springer, 2020. doi:10.1007/978-3-030-42071-0_18.
- 61 Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009. doi:10.1007/978-3-642-04128-0_51.