Karlsruhe Institute of Technology

Parameterized Complexity of Zero Forcing and its Variants

Bachelor's Thesis of

Lennard Hofmann

At the Department of Informatics
Institute of Theoretical Informatics (ITI)

Reviewer:	TT-Prof. Dr. Thomas Bläsius
Second reviewer:	Prof. Dr.-Ing. Marvin Künnemann
Advisor:	Max Göttlicher

December 12, 2025 – April 13, 2026

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, April 13, 2026

. . . .

(Lennard Hofmann)

Abstract

The zero forcing number of a graph is the minimum number of vertices that must be infected such that all other vertices become infected after iterated application of the following rule: a vertex becomes infected when it is the only uninfected neighbor of an adjacent infected vertex. We show that the zero forcing number can be computed in linear time on tree-cographs and P_4-sparse graphs. The proof is based on a new formula for the zero forcing number of joins of graphs. The remainder of the thesis presents hardness results for related problems, including Connected Forcing, which requires that the initial set of infected vertices induces a connected subgraph, and k-Forcing, where vertices can infect up to k neighbors. It is known that Connected Forcing is fixed-parameter tractable when parameterized by solution size. We prove that it does not admit a polynomial kernel unless the polynomial hierarchy collapses. In addition, we determine the parameterized complexity of k-Forcing by showing that it is W[P]-complete.

Zusammenfassung

Die Zero-Forcing-Zahl eines Graphen ist die Minimalzahl Knoten, die infiziert werden müssen, damit alle anderen Knoten durch iterierte Anwendung folgender Regel infiziert werden: Ein Knoten wird infiziert, wenn er der einzige nicht infizierte Nachbar eines adjazenten infizierten Knotens ist. Wir zeigen, dass die Zero-Forcing-Zahl in Linearzeit auf Tree-Cographs und P_4-Sparse Graphs berechnet werden kann. Der Beweis basiert auf einer neuen Formel für die Zero-Forcing-Zahl des Joins zweier Graphen. Außerdem untersuchen wir die parametrisierte Komplexität von verwandten Problemen, darunter Connected Forcing, bei dem die anfänglich infizierten Knoten einen zusammenhängenden Teilgraphen induzieren müssen, und k-Forcing, bei dem infizierte Knoten bis zu k Nachbarn infizieren können. Es ist bekannt, dass Connected Forcing in FPT-Zeit gelöst werden kann, wenn es nach Lösungsgröße parametrisiert wird. Wir zeigen, dass es keinen polynomialen Kernel besitzt, es sei denn, die Polynomialzeithierarchie kollabiert. Des Weiteren bestimmen wir die parametrisierte Komplexität von k-Forcing, indem wir beweisen, dass es W[P]-vollständig ist.

Contents

1 Introduction

The zero forcing number was introduced in 2008 by the AIM Minimum Rank Special Graphs Work Group [AIM08] as an upper bound for maximum nullity, a graph parameter from linear algebra that is notoriously hard to compute — even on small graphs with less than ten vertices [BS19 | BHHS25]. Since then, Zero Forcing has developed into an active research area in graph theory, with MathSciNet listing over 240 works citing the original paper. Zero Forcing was used to model phase measurement units in electric power networks [HHHH02] and has been independently studied by physicists for controlling quantum systems [BG07]. It has also been shown to be dual to Z-Grundy Domination [Bre+17], and proven equivalent to the pursuit-evasion game *fast–mixed searching* [FMY16, Theorem 3.1].

Zero Forcing Definition Let G be a (finite simple undirected) graph and S a set of vertices. Color all vertices in S black and the remaining vertices white. We say that S is a *zero forcing set* if every vertex of G can be turned black by repeatedly applying the *color change rule*: a black vertex with exactly one white neighbor turns that neighbor black. The *zero forcing number* $Z(G)$ is the minimum size of a zero forcing set. Clearly, $1 \leq Z(G) \leq |G|$, and the lower bound is tight for paths only, while the upper bound is attained only by graphs without edges. The graphs with $Z(G) = 2$ and $Z(G) \geq |G| - 2$ have been characterized in [Row11b, Theorem 2.2.3] and [AIM08, Theorem 5.4]. Given an integer d and a graph G, the Zero Forcing problem is to decide whether $Z(G) \leq d$.

Variants of Zero Forcing The Power Dominating Set problem combines Dominating Set with Zero Forcing by asking for a minimum set of vertices whose closed neighborhood is a zero forcing set [HHHH02]. It has been shown to be W[P]-complete [BG25]. Another variant is k-Forcing [ACDP15], which is obtained by relaxing the color change rule in the following way: a black vertex with at most k white neighbors turns all of them black. Its parameterized complexity has not been studied thus far; we show that it is W[P]-complete for $k \geq 2$ using a reduction similar to the one for Power Dominating Set. In addition, we show that Zero Forcing already becomes W[P]-complete when allowing for some vertices to be precolored in black. Previously, this variant (called Precolored Zero Forcing) was only known to be W[2]-hard [Caz+19, Theorem 2]. In some sense, Precolored Zero Forcing encapsulates the fundamental reason why k-Forcing and Power Dominating Set are W[P]-hard.

Parameterized Complexity Classical complexity theory studies running time as a function of the input size, such as the number of vertices in a graph or variables in a SAT instance. In practice, input size alone often fails to predict difficulty, as SAT solvers may require vastly different running times on instances of equal size. Parameterized complexity theory accounts for this by assigning each instance an integer (the *parameter*) that captures its difficulty, and by identifying NP-hard problems that are solvable in polynomial time for small parameter values. See [Cyg+15] for a full introduction to parameterized complexity.

1.1 Related Work

Computing the zero forcing number of a graph is NP-hard [Yan13, Corollary 6.6], but there exist linear-time algorithms for trees [FH07, Algorithm 2.3], cacti [Yan13, Corollary 7.6], proper interval graphs [Yan13, Corollary 7.8][Caz+19, Theorem 4], and cographs [Kos19, Theorem 3.70] as well as a polynomial-time algorithm for block graphs (graphs whose biconnected components are cliques) [AM25]. We extend the algorithm of Kos to P_4-sparse graphs and tree-cographs.

The Zero Forcing problem can be solved in time $2^{\mathcal{O}(w^2)} \cdot |G|$, where w is the treewidth of G [Sch26]. Thus, Zero Forcing is FPT when parameterized by the zero forcing number $Z(G)$ because $w \le Z(G)$ holds for any graph. Although this runtime is impractical even for very small values of w, it implies that Zero Forcing can be solved in linear time on graph classes of bounded treewidth such as trees, outerplanar graphs, and cacti. All of the above also holds true for the related problem Connected Forcing, where the initial set of black vertices must induce a connected subgraph. Therefore, it is natural to ask whether these problems admit a polynomial kernel. For Connected Forcing, we settle this question in the negative using a modified reduction from Red-Blue Dominating Set to Connected Vertex Cover by Dom et al. [DLS14].

There exist formulas for the zero forcing number of graphs built from smaller graphs using operations like the Cartesian product [Cam+26], strong product, tensor product [HCY10, Lemma 17], corona product [JIBR16 | Cam+26], join, and vertex-sum [Tak13]. We improve the join formula of Taklimi by generalizing it to arbitrary graphs and extending it to Connected Forcing.

An online catalog of zero forcing numbers for various graph families can be found in [Hog+]. There exists software to efficiently compute zero forcing numbers of medium-sized graphs.[1] For a more comprehensive survey on zero forcing and its variants, we refer the reader to the monograph “Inverse Problems and Zero Forcing for Graphs” [HLS22].

1.1.1 Relations to Other Graph Parameters

Let G be a connected graph on $n \ge 2$ vertices. Then

$$\delta \le \mathrm{dg}(G) \le \mathrm{tw}(G) \le \mathrm{pw}(G) \le \mathrm{ppw}(G) \le \mathrm{Z}(G) \le \mathrm{ml}(G) \le n - 1,$$

where δ, $\mathrm{dg}(G)$, $\mathrm{tw}(G)$, $\mathrm{pw}(G)$, $\mathrm{ppw}(G)$, and $\mathrm{ml}(G)$ are the minimum degree, degeneracy, treewidth, pathwidth, proper pathwidth, and max leaf number of G, respectively. The parameters treewidth and (proper) pathwidth each have multiple equivalent definitions. Using the clique number $\omega(G)$ (the size of a maximum clique in G), they are characterized by

$$\mathrm{dg}(G) = \max\{\delta(H) \mid H \text{ is a subgraph of } G\}, \tag{1}$$
$$\mathrm{tw}(G) + 1 = \min\{\omega(H) \mid H \text{ is a supergraph of } G \text{ and chordal}\}, \tag{2}$$
$$\mathrm{pw}(G) + 1 = \min\{\omega(H) \mid H \text{ is a supergraph of } G \text{ and an interval graph}\}, \tag{3}$$
$$\mathrm{ppw}(G) = \min\{Z(H) \mid H \text{ is a supergraph of } G\}, \tag{4}$$

[1] https://pypi.org/project/zftools/

where the first three equations are well-known [Bod98] and equation (4) follows from a theorem by Barioli et al. [Bar+13, Theorem 2.39][HLS22, Corollary 9.130]. The following parameter is sometimes also called proper pathwidth

$$\mathrm{bw}(G) + 1 = \min\{\omega(H) \mid H \text{ is a supergraph of } G \text{ and a } \textit{proper} \text{ interval graph}\} \tag{5}$$

but is better known as *bandwidth* and distinct from proper pathwidth [KS96, Theorem 3.2].

A k*-tree* is a graph that can be reduced to K_k by repeatedly removing a simplicial vertex of degree k. Note that 1-trees are the same as trees and that the subgraphs of k-trees are precisely the graphs of treewidth at most k [Bod98, Theorem 35]. Similarly, we can define $\mathrm{ppw}(G)$ as the minimum integer k such that G is a subgraph of K_{k+1} or a k-tree with exactly two vertices of degree k [TUK95, Theorem 2]. We note in passing that these k-trees (known as *linear* k*-trees* or k*-paths*) have minimum degree $\delta = Z(G) = k$ [Yan13, Corollary 7.9][Bar+13, Lemma 2.37].

Whereas $\mathrm{pw}(G)$ and $\mathrm{ppw}(G)$ may differ by at most one [TUK95, Theorem 1], the zero forcing number $Z(G)$ can be arbitrarily far apart from $\mathrm{ppw}(G)$. For example, the star $K_{1,n-1}$, $n \geq 4$, has zero forcing number $n - 2$, but its proper pathwidth is only 2 since it is a subgraph of the fan graph $P_{n-1} \vee K_1$, which is obtained from the path on $n - 1$ vertices by adding a universal vertex.

The *max leaf number* $\mathrm{ml}(G)$ is the maximum number of leaves in a spanning tree of G. For $|G| \geq 3$, it satisfies $\mathrm{ml}(G) + \gamma_c(G) = |G|$, where $\gamma_c(G)$ is the minimum size of a dominating set that induces a connected subgraph [HL84, Theorem 2]. The inequality $Z(G) \leq \mathrm{ml}(G)$ was discovered by Amos et al. [ACDP15, Corollary 4.3].

A *forcing chain* is a sequence $(v_1, \dots, v_k)$ of vertices such that v_i turns v_{i+1} black for $1 \leq i < k$. It is easy to see that any vertex of a zero forcing set is the start of a forcing chain, and that these forcing chains partition the vertex set into induced paths. Such a partition is known as a *path cover*, and the minimum number of paths in a path cover is the *path cover number* $P(G)$. Thus, $P(G) \leq Z(G)$ holds [Owe09, Corollary 4.3]. This inequality is sharp when G is a cactus [Row11a].

Recall that zero forcing was originally introduced to bound the maximum nullity of a graph, which is defined as follows. For a graph $G = (V, E)$ with $V = \{1, \dots, n\}$, consider the family $\mathcal{S}(G)$ of all real symmetric matrices $A \in \mathbb{R}^{n\times n}$ whose off-diagonal entries A_{ij}, $i \neq j$, are nonzero if and only if $ij \in E$. The *maximum nullity* $M(G)$ of G is the maximum nullity of those matrices, that is,

$$M(G) = \max\{\dim(\ker A) \mid A \in \mathcal{S}(G)\}.$$

Maximum nullity $M(G)$ is dual to minimum rank $\mathrm{mr}(G)$ in the sense that $M(G) + \mathrm{mr}(G) = |G|$ holds for any graph G. Roughly speaking, the inequality $Z(G) \geq M(G)$ relates to vertices turning black corresponding to coordinates of a vector in the kernel being forced to be zero; hence the name "zero forcing". [AIM08]

There also exist inequalities relating zero forcing number with number of leaves [WWZ20], independence number [Bre+17, Proposition 3.2], girth [FR20], maximum degree [CLTC26], spectral radius [ZWWJ22], metric dimension [EKY17], matching number [JZJ23], and general position number [HHK21].

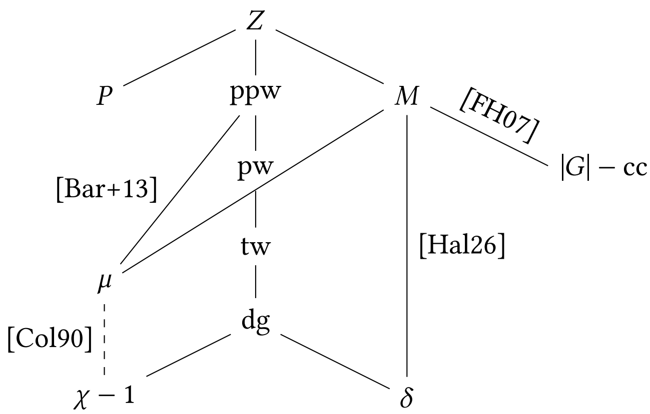


Figure 1.1: Hasse diagram of zero forcing lower bounds, where $\chi - 1$ is chromatic number minus 1, cc is the intersection number (the minimum number of cliques needed to cover all edges), and μ is Colin de Verdière's invariant. The inequality $\chi - 1 \leq \mu$ was conjectured by Colin de Verdière in 1990 and remains wide open.

1.2 Outline

This thesis is organized as follows. In the next chapter we introduce our terminology and notation. In Chapter 3 we derive formulas to compute the (connected) zero forcing number of joins of graphs. In Chapter 4, we provide formulas for the zero forcing number of spiders and the connected zero forcing number of tree complement graphs. These formulas form the basis for linear-time algorithms to compute the zero forcing number on P_4-sparse graphs and tree-cographs. In Chapter 5 we prove that Connected Forcing does not admit a polynomial kernel unless the polynomial hierarchy collapses, thus resolving a question posed by Scheffler [Sch26]. In Chapter 6 we determine the parameterized complexity of the problems k-Forcing and Precolored Zero Forcing by showing that they are W[P]-complete. We conclude with open problems. The Appendix contains a full list of computational problems considered in this thesis.

2 Preliminaries

2.1 Graph Theory

We use standard graph theory notation and refer to [Die25] for terms not defined here. All graphs in this thesis are *simple* (without loops) and contain at least one vertex. Consider a graph $G = (V, E)$ with vertex set $V(G) \coloneqq V$ and edge set $E(G) \coloneqq E$. For an edge $\{u, v\} \in E$ between two vertices u and v we simply write uv, and the number of vertices in G is denoted $|G| \coloneqq |V(G)|$. The subgraph $G[S]$ *induced* by a set $S \subseteq V$ is the graph $(S, \{uv \mid u, v \in S,\ uv \in E\})$. Deleting a vertex v results in the induced subgraph $G - v \coloneqq G[V \setminus \{v\}]$. Subdividing an edge vw means replacing it with two edges vs and sw, where s is a new vertex called the *subdivision vertex*.

The *neighborhood* $N(v) \coloneqq \{u \in V \mid uv \in E\}$ of a vertex v is the set of vertices adjacent to v. A vertex is *simplicial* if its neighborhood forms a clique, that is, all neighbors are pairwise adjacent. The *degree* of a vertex is the cardinality of its neighborhood. A vertex of degree 0 is an *isolated vertex* and a vertex of degree 1 is a *leaf* (plural *leaves*). We say that a graph without isolated vertices is *isolate-free*. The *closed neighborhood* of a vertex v is defined as $N[v] \coloneqq N(v) \cup \{v\}$. The closed neighborhood of a set $S \subseteq V$ is the union $N[S] \coloneqq \bigcup_{v \in S} N[v]$.

A set $D \subseteq V$ of vertices is a *dominating set* if every vertex $v \in V \setminus D$ has a neighbor in D. Equivalently, $D \subseteq V$ is a dominating set if $N[D] = V$. A vertex v is a *universal vertex* if $\{v\}$ is a dominating set. We say that a set $S \subseteq V$ *dominates* its closed neighborhood and that it *totally dominates* its (open) neighborhood $N(S) \coloneqq \bigcup_{v \in S} N(v)$.

Given a graph $G = (V, E)$, its *complement* is the graph $\overline{G} = (V, \{uv \mid u, v \in V,\ uv \notin E\})$. The *empty graph* nK_1 is the graph with n vertices and no edges. Its complement is the *clique* K_n. A set $I \subseteq V$ of vertices is an *independent set* if I induces an empty subgraph. A *2-coloring* of G is a partition of the vertex set V into two independent sets A and B. We say that a graph is *bipartite* if it admits a 2-coloring.

Let G and H be graphs with disjoint vertex sets. Their *union* is the graph $G \sqcup H$ with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. The *join* of G and H is the graph $G \vee H$ obtained by adding the edges $\{uv \mid u \in V(G), v \in V(H)\}$ to their union $G \sqcup H$. As the join operation is commutative and associative, we omit parentheses in $(G_1 \vee G_2) \vee G_3$ and write $\bigvee_{i=1}^{3} G_i$ instead. The *complete bipartite graph* $K_{m,n}$ is the join of two empty graphs, that is, $K_{m,n} = mK_1 \vee nK_1$.

A graph G is *connected* if there is a path between every pair of vertices. Otherwise, we say that the graph is *disconnected* and refer to its maximal connected subgraphs as *components*. A tree T is a connected acyclic graph. If one vertex of T is designated as the root, then T is referred to as a *rooted tree*. Given two distinct vertices $u, v \in V(T)$, we say that u is a *descendant* of v if v is contained in the unique path from u to the root of T. In particular, every non-root vertex is a descendant of the root.

Let D be a directed graph. If D contains an edge (s, t) from s to t, then s is an *in-neighbor* of t and t is an *out-neighbor* of s. If s has no in-neighbors (resp. out-neighbors), we say that s is a *source* (resp. *sink*). The number of out-neighbors of s is denoted $\operatorname{outdeg}(s)$. An *out-tree* is a directed graph whose underlying undirected graph is a tree and whose edges are oriented away from the root. A *binary* out-tree is one where every vertex has at most two out-neighbors.

2.2 Zero Forcing

Let G be a graph (resp. a directed graph). A set $S \subseteq V(G)$ of vertices is a *forcing set* if all vertices of G are black after the following iterated process:

- Initially, color all vertices of S black and the rest white.
- Whenever a black vertex b has exactly one white (out-)neighbor w, color w black. We say that b *forces* w and write $b \to w$.

If vertex x forces y and y subsequently forces z, then we say that x *transitively forces* z. This process stops when no vertex can force another. At any point, there might be multiple vertices that can force a neighbor, but it does not matter which of these forces is performed; indeed, the final set of black vertices only depends on S. Clearly, $V(G)$ is a forcing set, so it is natural to define the *forcing number* $Z(G)$ as the minimum size of a forcing set. Traditionally, forcing sets are called *zero forcing sets*, and the forcing number is correspondingly known as *zero forcing number*.

Observation 2.1: *If G is a disconnected graph with k components $G_1, \dots, G_k$, then*

$$Z(G) = \sum_{i=1}^{k} Z(G_i).$$

Proposition 2.2: *Consider the path graph P_n on $n \in \mathbb{N}$ vertices. A set S of vertices is a forcing set of P_n if and only if S contains a leaf or a pair of adjacent vertices.*

Proof. We can linearly order the vertices $v_1, \dots, v_n$ of P_n such that $v_i v_j$ is an edge if and only if $j = i + 1$. Clearly, the leaves v_1 and v_n each can transitively force the entire path. Also, for any edge $v_i v_{i+1}$ with both endpoints black, the left endpoint v_i can transitively force every vertex v_j $(j < i)$ on its left, while the right endpoint v_{i+1} can transitively force every vertex v_j $(j > i + 1)$ on its right. Thus, the set $\{v_i, v_{i+1}\}$ and its supersets are forcing sets. Conversely, if S contains neither a leaf nor a pair of adjacent vertices, then every vertex in S has two white neighbors at the start of the forcing process, so no force can be made. ∎

Figure 2.1: Two forcing sets of the path P_4

A forcing set is said to be *total* (*connected*) if it induces an isolate-free (connected) subgraph. If G is an isolate-free (connected) graph, then the *total (connected) forcing number* $Z_t(G)$ ($Z_c(G)$) is the minimum size of a total (connected) forcing set. For example, $Z(P_n) = Z_c(P_n) = 1$, but $Z_t(P_n) = 2$, where P_n is a path on $n \geq 2$ vertices. Let G be a connected graph that is not a path. Then $Z(G) \geq 2$, so any connected forcing set of G is also a total forcing set and $Z(G) \leq Z_t(G) \leq Z_c(G)$.

2.3 Parameterized Complexity

A *parameterized problem* Π is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. To *solve* a parameterized problem Π means being able to decide for every instance $(x, d) \in \Sigma^* \times \mathbb{N}$ whether $(x, d) \in \Pi$ holds. We say that Π is *fixed-parameter tractable* (FPT) if it can be solved in time $f(d)|x|^{\mathcal{O}(1)}$ for some computable function f that only depends on the parameter d.

In classical complexity theory, one can rule out the existence of a polynomial-time algorithm by giving a Karp reduction from an NP-hard problem (assuming P ≠ NP). Similarly, we can show that a parameterized problem Π' is not FPT by giving an fpt-reduction from another parameterized problem Π known to be not FPT. Here, an *fpt-reduction* is an algorithm with the following property: For any given instance (x, d) of Π, it outputs in FPT time $f(d)|x|^{\mathcal{O}(1)}$ an instance (x', d') of Π' such that

$$[(x, d) \in \Pi \iff (x', d') \in \Pi'] \text{ and } d' \leq g(d)$$

where f and g are computable functions.

The class of fixed-parameter tractable problems sits at the bottom of the W-hierarchy

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \text{W}[3] \subseteq \dots,$$

where all inclusions are believed to be proper. The classes W[1], W[2], etc. are defined to contain those parameterized problems that are fpt-reducible to certain circuit satisfiability problems of increasing complexity. If C is a class of parameterized problems, then we say that $\Pi' \in C$ is *C-complete* if every problem in C has an fpt-reduction to Π'. For example, the Dominating Set problem is W[2]-complete and W[2] may also be defined as the class of parameterized problems fpt-reducible to Dominating Set. A common way to argue that a parameterized problem is probably not FPT is to give an fpt-reduction from a W[t]-complete problem ($t \geq 1$).

See Chapter 5 for definitions related to kernelization.

3 Grundy Domination and Joins of Graphs

In this chapter, we derive a formula for the forcing number $Z(G \vee H)$ of the join of two graphs G and H with disjoint vertex sets (recall that the join $G \vee H$ has vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H) \cup \{uv \mid u \in V(G), v \in V(H)\}$). Its proof uses the language of Z-Grundy domination, defined below.

Let G be a graph. A sequence $(s_1, \ldots, s_k)$ of vertices is called a *Z-sequence* if each s_i totally dominates some vertex that was not dominated by the previous vertices, that is,

$$N(s_i) \setminus \bigcup_{j=1}^{i-1} N[s_j] \neq \emptyset.$$

We say that s_i *footprints* the vertices $N(s_i) \setminus \bigcup_{j=1}^{i-1} N[s_j]$. The maximum length of a Z-sequence is the *Z-Grundy domination number* $\mathrm{mz}(G)$, which is often denoted $\gamma_{\mathrm{gr}}^{Z}(G)$ in the literature.

Z-sequences were introduced by Brešar et al. [Bre+17], who also showed $Z(G) + \mathrm{mz}(G) = |G|$ for any graph G. In particular, observe that $Z(G) + \mathrm{mz}(G) = |G|$ remains true when adding an isolated vertex to G. This is because an isolated vertex is contained in every forcing set, but it is not part of any Z-sequence as it cannot footprint any vertex. More strongly, Brešar et al. proved the following theorem (but they stated it only for isolate-free graphs).

Theorem 3.1 ([Bre+17, Theorem 2.2]): *Suppose that $(s_1, \ldots, s_k)$ is a Z-sequence of a graph G. Its complement $V(G) \setminus \{s_1, \ldots, s_k\}$ is a forcing set of G. Conversely, the complement of any forcing set forms a Z-sequence.*

Now, we are ready to prove the main result of this chapter.

Theorem 3.2: *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs such that V_1 and V_2 are disjoint. Then the join $G = G_1 \vee G_2$ satisfies*

$$\mathrm{mz}(G) = \max\{\mathrm{mz}(G_1) + \eta(G_1),\ \ \mathrm{mz}(G_2) + \eta(G_2)\},$$

where $\eta(H) = \min\{2,\ n_0(H)\}$ and $n_0(H)$ is the number of isolated vertices in H.

Before proving this, we remark that a similar (but incorrect) identity can be found in the Ph.D. thesis of Tim Kos [Kos19, Proposition 3.38]. It uses $\min\{1,\ n_0(H)\}$ instead of $\min\{2,\ n_0(H)\}$ for $\eta(H)$, which leads to contradictions such as $1 = \mathrm{mz}(2K_1 \vee K_1) = \mathrm{mz}(P_3) = 2$. Our proof of Theorem 3.2 is inspired by Kos.

Proof. First, we show $\mathrm{mz}(G) \geq \mathrm{mz}(G_p) + \eta(G_p)$ for all $p \in \{0, 1\}$. Consider a Z-sequence $S = (s_1, s_2, \ldots, s_k)$ of length $k = \mathrm{mz}(G_p)$ in G_p. Clearly, S is also a Z-sequence in G; thus $\mathrm{mz}(G) \geq \mathrm{mz}(G_p)$. If G_p has an isolated vertex v, then $(v, s_1, \ldots s_k)$ is a Z-sequence in G since v footprints all of V_{1-p} but none of V_p. Hence, $\mathrm{mz}(G) \geq \mathrm{mz}(G_p) + 1$. Now, suppose that G_p has two isolated vertices u and v (and possibly more). Then $(v, s_1, \ldots s_k, z)$ is a Z-sequence in G for any $z \in V_{1-p}$ as z footprints u. Thus, $\mathrm{mz}(G) \geq \mathrm{mz}(G_p) + 2 = \mathrm{mz}(G_p) + \eta(G_p)$.

Next, we show that no longer Z-sequence exists. Let $S = (s_1, s_2, \dots, s_k)$ be a Z-sequence in G of maximum length k. By renaming G_1 and G_2 if necessary, we may assume $s_1 \in V_1$. We show $k \leq \mathrm{mz}(G_1) + \eta(G_1)$, from which the theorem follows.

Case 1 (S contains no vertex from G_2): If s_1 is not isolated in G_1, then S is also a Z-sequence in G_1. Therefore, $k \leq \mathrm{mz}(G_1)$ and we are done. Otherwise, s_1 is isolated in G_1 and $\eta(G_1) \geq 1$. Notice that s_1 footprints all of V_2, so $(s_2, \dots, s_k)$ must be a Z-sequence in G_1. This implies $k \leq \mathrm{mz}(G_1) + 1 \leq \mathrm{mz}(G_1) + \eta(G_1)$.

Case 2 (S contains a vertex from G_2): Consider the minimum index i such that $s_i \in V_2$. Since $N(s_1) \cup N(s_i) = V(G)$, the vertex s_i must be the last vertex in S, that is, $i = k$. In particular, s_k is the only vertex from G_2 in S. Notice that s_1 footprints all of V_2, so $(s_2, \dots, s_{k-1})$ must be a Z-sequence in G_1. Hence, $k \leq \mathrm{mz}(G_1) + 2$ and we are done if $\eta(G_1) = 2$. Suppose not, that is, $\eta(G_1) \leq 1$. Let $f \in V_1$ be a vertex footprinted by s_k. If f is isolated in G_1, then s_1 is not isolated in G_1, so $(s_1, \dots, s_{k-1})$ must be a Z-sequence in G_1. Thus, $k \leq \mathrm{mz}(G_1) + 1 = \mathrm{mz}(G_1) + \eta(G_1)$. If f is not isolated in G_1, then we can take any neighbor $v \in V_1 \cap N(f)$ to obtain a Z-sequence $(s_2, \dots, s_{k-1}, v)$ of length $k - 1$ in G_1. Therefore, $k \leq \mathrm{mz}(G_1) + 1$ and we are done if $\eta(G_1) \geq 1$. Suppose $\eta(G_1) = 0$. Then s_1 is not isolated in G_1. Thus, $(s_1, \dots, s_{k-1}, v)$ is a Z-sequence in G_1, and $k \leq \mathrm{mz}(G_1)$. ∎

Recall that $Z(G) + \mathrm{mz}(G) = |G|$ holds for every graph by Theorem 3.1. This allows us to restate Theorem 3.2 in terms of the zero forcing number.

Corollary 3.3: *Let G_1 and G_2 be graphs with disjoint vertex sets. Then*

$$Z(G_1 \vee G_2) = \min\{|G_2| + Z(G_1) - \eta(G_1),\ \ |G_1| + Z(G_2) - \eta(G_2)\},$$

where $\eta(H) = \min\{2, n_0(H)\}$ and $n_0(H)$ is the number of isolated vertices in H.

Corollary 3.4: *Let K_r be a clique on r vertices and H be a graph. Then*

$$Z(K_r \vee H) = r + Z(H) - \eta(H), \tag{1}$$

where $\eta(H) = \min\{2, n_0(H)\}$ and $n_0(H)$ is the number of isolated vertices in H.

Proof. By Corollary 3.3, we have

$$Z(K_r \vee H) = \min\{|H| + r - 1,\ \ r + Z(H) - \eta(H)\}, \tag{2}$$

If H has no edges, then $Z(H) = |H|$ and $\eta(H) \geq 1$, so $r + Z(H) - \eta(H) \leq |H| + r - 1$. Otherwise, $Z(H) \leq |H| - 1$. In both cases, (2) simplifies to (1). ∎

The following observations about forcing sets in arbitrary graphs will be useful in the next section.

Observation 3.5 ([Bar+10, Lemma 2.8])**:** *If S is a minimum forcing set, then every non-isolated vertex $u \in S$ has a neighbor outside S.*

Proof. Suppose that S is a minimum forcing set with $N[u] \subseteq S$. Then we can remove any neighbor v of u from S to obtain a smaller forcing set as u can force v. ∎

Observation 3.6: *If v is a vertex adjacent to $k \geq 2$ leaves, then every forcing set contains at least $k - 1$ of these leaves, and v is not contained in any minimum forcing set.*

Proof. Let S be a forcing set. As v can only force one of its leaves, the others must be in S. Let S be a minimum forcing set and $u \in S$ a neighbor of v. By Observation 3.5, $v \notin S$. ∎

3.1 Total and Connected Forcing

A forcing set is said to be *total* (*connected*) if it induces an isolate-free (connected) subgraph. If G is an isolate-free (connected) graph, then the *total (connected) forcing number* $Z_t(G)$ ($Z_c(G)$) is the minimum size of a total (connected) forcing set. For example, $Z(P_n) = Z_c(P_n) = 1$, but $Z_t(P_n) = 2$, where P_n is a path on $n \geq 2$ vertices.

Using results from Taklimi [Tak13] and Davilla et al. [DHP23], Brimkov et al. [BDS25] showed that the join $G = G_1 \vee G_2$ of two isolate-free graphs G_1 and G_2 satisfies

$$Z(G) = Z_t(G) = Z_c(G) = \min\{|G_2| + Z(G_1),\, |G_1| + Z(G_2)\}.$$

In this section, we extend this result to the general case where G_1 and G_2 are arbitrary graphs. Our main theorem is the following.

Theorem 3.7: *Let G_1 and G_2 be graphs with disjoint vertex sets and $G = G_1 \vee G_2$ be the join of G_1 and G_2. If G is not a path, then:*

$$Z_t(G) = Z_c(G) = \min\{|G_2| + Z(G_1) - \eta(G_1, |G_2|),\, |G_1| + Z(G_2) - \eta(G_2, |G_1|)\}, \qquad (3)$$

where $\eta(H, x) = \min\{2,\, n_0(H),\, x\}$ and $n_0(H)$ is the number of isolated vertices in H.

Notice that the right-hand side of (3) is identical to the one in Corollary 3.3 when $|G_1| \geq 2$ and $|G_2| \geq 2$. For the case $G_2 = K_1$, the next two propositions provide even simpler formulas.

Proposition 3.8 ([Fer+25, Proposition 3.5]): *Let H be a graph and $G = H \vee K_1$ be the graph obtained by adding a universal vertex u to H.*

1. *If H is isolate-free, then $Z(G) = Z(H) + 1$.*
2. *If H has exactly one isolated vertex, then $Z(G) = Z(H)$.*
3. *If H has at least two isolated vertices, then $Z(G) = Z(H) - 1$.*

Proof. Follows immediately from Corollary 3.4. ∎

In general, deleting a vertex u from a graph G changes the zero forcing number by at most 1, that is, $-1 \leq Z(G) - Z(G - u) \leq 1$ [Owe09, Theorem 6.4]. The above proposition shows that both bounds are tight. Next, we obtain equations for total and connected forcing numbers.

Proposition 3.9: *Suppose that H is a graph and that $G := H \vee K_1$ is not a path.*

1. *If H is isolate-free, then $Z_t(G) = Z_c(G) = Z(H) + 1$.*
2. *Otherwise, $Z_t(G) = Z_c(G) = Z(H)$.*

Proof. *Case 1* (H is isolate-free): By [HLS22, Proposition 9.16], we have $Z(G) = Z(H) + 1$ and there exists a minimum zero forcing set S of G such that $u \in S$. Since u is adjacent to every vertex of H, the induced subgraph $G[S]$ is connected. Thus, S is also a connected forcing set of G. Since G is not a path, this implies $Z(G) = Z_t(G) = Z_c(G)$.

Case 2 (H has an isolated vertex v): Let S be a minimum forcing set of H and consider a sequence of forces F that turns the rest of H black. Since v is isolated, $v \in S$. We claim that $S' := (S \cup \{u\}) \setminus \{v\}$ is a connected forcing set of G. Color all vertices in S' black. Since $u \in S'$, the induced subgraph $G[S']$ is connected, and every vertex of G (except v) can be turned black using the forces in F. After these forces have been made, v is the only white vertex, so it can be forced by u. Since G is not a path, this implies

$$Z(G) \leq Z_t(G) \leq Z_c(G) \leq |S'| = |S| = Z(H). \tag{4}$$

If v is the only isolated vertex of H, then $Z(G) = Z(H)$ by Proposition 3.8, so we have equality in (4). Suppose now that H has at least two isolated vertices. Let S be a minimum zero forcing set of G. Since the universal vertex u is adjacent to at least two leaves in G, one of the leaves must be in S and $u \notin S$ by Observation 3.6. Therefore, this leaf is isolated in $G[S]$ and S is not a total forcing set. Hence,

$$Z_t(G) \geq Z(G) + 1 \geq Z(G - u) = Z(H)$$

because deleting u from G increases the forcing number by at most 1. Using (4), we conclude

$$Z(G) + 1 = Z_t(G) = Z_c(G) = Z(H).$$ ∎

Recall Theorem 3.7.

Theorem 3.7: *Let G_1 and G_2 be graphs with disjoint vertex sets and $G = G_1 \vee G_2$ be the join of G_1 and G_2. If G is not a path, then:*

$$Z_t(G) = Z_c(G) = \min\{|G_2| + Z(G_1) - \eta(G_1, |G_2|),\ |G_1| + Z(G_2) - \eta(G_2, |G_1|)\}, \tag{3}$$

where $\eta(H, x) = \min\{2,\ n_0(H),\ x\}$ and $n_0(H)$ is the number of isolated vertices in H.

Proof. We distinguish two cases.

Case 1 ($|G_1| \geq 2$ and $|G_2| \geq 2$): Then $\eta(H, x) = \min\{2, n_0(H)\}$ for $x \in \{|G_1|, |G_2|\}$ and (3) simplifies to $Z(G) = Z_t(G) = Z_c(G)$ by Corollary 3.3. To show equality in $Z(G) \leq Z_t(G) \leq Z_c(G)$, we prove that G has a connected forcing set of size $Z(G)$. Consider a Z-sequence $S = (s_1, \ldots, s_k)$ in G of length $k = \mathrm{mz}(G)$. By renaming G_1 and G_2 if necessary, we may assume $s_1 \in V(G_1)$. Recall that S contains at most one vertex from G_2 and notice that S cannot contain all vertices of G_1 (if it did, the last vertex would not footprint anything). Hence, the complement $\overline{S} := V(G) \setminus \{s_1, \ldots, s_k\}$ intersects $V(G_1)$ and $V(G_2)$. Thus, $G[\overline{S}]$ is connected. By Theorem 3.1, $\overline{S}$ is a minimum forcing set of G. Therefore, $\overline{S}$ is a connected forcing set of size $|G| - \mathrm{mz}(G) = Z(G)$.

Case 2 ($|G_1| = 1$ or $|G_2| = 1$): Without loss of generality, assume $|G_1| = 1$. We need to show

$$Z_t(G) = Z_c(G) = \min\{|G_2|,\ 1 + Z(G_2) - \eta(G_2, 1)\}. \tag{5}$$

If G_2 has no edges, then $Z(G_2) = |G_2|$ and $\eta(G_2, 1) = 1$, so $1 + Z(G_2) - \eta(G_2, 1) = |G_2|$. If G_2 has at least one edge, then $Z(G_2) \leq |G_2| - 1$. In both cases, (5) simplifies to

$$Z_t(G) = Z_c(G) = 1 + Z(G_2) - \eta(G_2, 1),$$

where $\eta(G_2, 1)$ is 0 if G_2 is isolate-free and 1 otherwise. Hence, the statement follows from Proposition 3.9. ∎

4 Forcing on Cographs and P_4-Sparse Graphs

As already observed by Tim Kos [Kos19, Theorem 3.70], the formulas of Theorem 3.2 and Theorem 3.7 show that the forcing numbers $Z(G)$, $Z_t(G)$ and $Z_c(G)$ of a cograph G can be computed in linear time using a straightforward recursion. In this chapter, we prove that $Z(G)$ can be computed in linear time on P_4-sparse graphs, a natural generalization of cographs. Even more generally, we say that a graph is a (q,t) *graph* if each set of at most q vertices induces at most t distinct P_4 subgraphs. In this notation, the cographs coincide with $(4,0)$ graphs and P_4-sparse graphs are exactly the $(5,1)$ graphs [BO95]. By definition, the class of $(5,1)$ graphs is hereditary (closed under taking induced subgraphs) and self-complementary since $\overline{P_4} = P_4$.

A non-standard definition of spiders, which play a key role in the characterization of P_4-sparse graphs, is presented next. Let H be a graph and $r \geq 2$ an integer. The *thin spider* with r legs and head H is the graph obtained by attaching a leaf to every vertex of the join $K_r \vee H$ that belongs to the clique K_r. The vertices of K_r are called *hubs* and the leaves attached to it are called *legs*.

Similarly, the *thick spider* with r legs and head H is the graph G obtained from $(K_r \vee H) \sqcup rK_1$ by adding the complement of a perfect matching between K_r and rK_1. The vertices of K_r are called *hubs* and the vertices of rK_1 are called *legs*. By construction, each hub has a unique leg that is not adjacent to it, and vice versa. It is easy to see that the complement of G is a thin spider with head $\overline{H}$ whose legs are the hubs of G. This is illustrated in Figure 4.1 below.

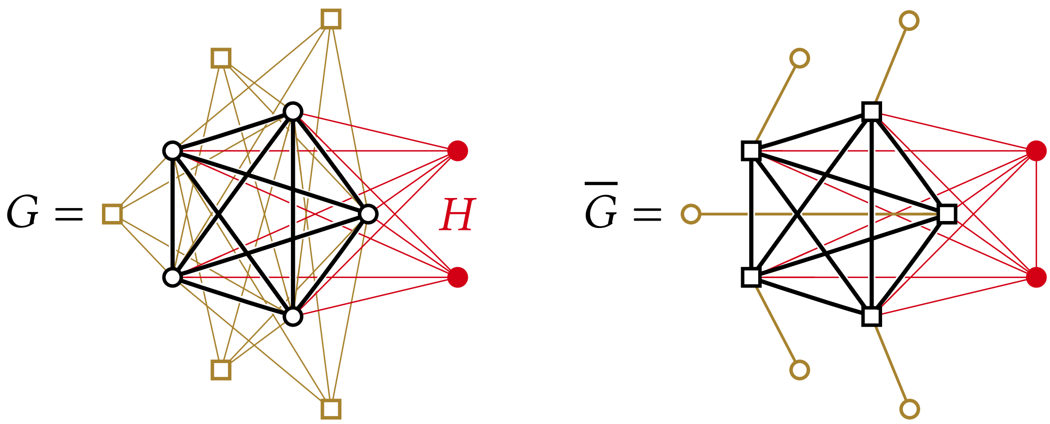


Figure 4.1: A thick spider with five legs and head $H = 2K_1$. Its complement $\overline{G}$ is a thin spider.

In this chapter, we repeatedly use the following two elementary zero forcing results.

Lemma 4.1: *Let F be a forcing set of a graph G. For any subset $S \subseteq F$, the graph G' obtained by attaching a leaf ℓ_i to every vertex $s_i \in S$ has forcing number at most $|F|$. Moreover, the set $F' := \{\ell_i \mid s_i \in S\} \cup (F \setminus S)$ of size $|F|$ is a forcing set of G'.*

Proof. Color all vertices of F' black. Every leaf ℓ_i can force its neighbor s_i. Now, F and the new vertices $V(G') \setminus V(G)$ are black, so the forcing process can continue as in G. ∎

Proposition 4.2 ([DY26, Theorem 6.5]): *If v is a simplicial vertex of a graph G, then*

$$Z(G - v) \leq Z(G).$$

Proof. Take a minimum forcing set S of G, color its vertices black, and consider a sequence of forces that turns the rest of G black. We need to show that $G - v$ has a forcing set of size $|S|$. If $v \in S$ and v does not force, then $S \setminus \{v\}$ is clearly a forcing set of $G - v$. If $v \in S$ and v forces w, then $(S \setminus \{v\}) \cup \{w\}$ is a forcing set of $G - v$. Lastly, if $v \notin S$, then v does not force and S is a forcing set of $G - v$. To see this, suppose that $x \to v \to y$ is a forcing chain, where $x, y \in N(v)$ are distinct neighbors of v. When v gets forced, it is the only white neighbor of x. In particular, y is black because $N(v)$ is a clique. Thus, the force $v \to y$ cannot occur. ∎

The zero forcing number of thin spiders is determined next.

Proposition 4.3: *Let G be a thin spider with $r \geq 2$ legs and head H. Then*

$$Z(G) = r + Z(H) - \eta(H),$$

where $\eta(H) = \min\{2, n_0(H)\}$ and $n_0(H)$ is the number of isolated vertices in H.

Proof. Consider the body $B := K_r \vee H$ of the spider G. Using Corollary 3.4 and Proposition 4.2 we have $r + Z(H) - \eta(H) = Z(B) \leq Z(G)$. It remains to show $Z(G) \leq r + Z(H) - \eta(H)$.

Let S_H be a minimum forcing set of H and let L be the set of legs in G.

Case 1 ($\eta(H) = 0$): Clearly, $L \cup S_H$ is a forcing set for G of size $r + Z(H)$.

Case 2 ($\eta(H) = 1$): Let v be the isolated vertex of H. Then the set $S_B := (V(K_r) \cup S_H) \setminus \{v\}$ has size $r + Z(H) - 1$ since $v \in S_H$. Further, S_B is a forcing set of B because v can be forced by any vertex from K_r after the rest of $V(H)$ has been forced. By Lemma 4.1, the graph G obtained by attaching leaves to $V(K_r) \subseteq S_B$ has forcing number at most $|S_B| = r + Z(H) - 1$.

Case 3 ($\eta(H) = 2$): Let v and w be isolated vertices of H and x be a vertex of K_r. Then the set $S_B := (V(K_r) \cup S_H) \setminus \{v, x\}$ has size $r + Z(H) - 2$. Further, S_B is a forcing set of B since w can force x, after which the forcing process can continue as in the previous case. We claim that $S_G := (L \cup S_H) \setminus \{v, \ell_x\}$, where ℓ_x is the leg attached to x, is a forcing set of G. Color all vertices of S_G black. The black legs can force every hub except x. Then, w can force x. Now that K_r is black, S_H can transitively force the rest of H except v. Then any vertex in $V(K_r) \setminus \{x\}$ can force v. Finally, x can force ℓ_x.

Therefore, we have $Z(G) \leq r + Z(H) - \eta(H)$ in every case. ∎

For the next results, we need some definitions. The *corona* $G \odot K_1$ of a graph G is the graph obtained by attaching a leaf to every vertex of G. For an integer $r \geq 2$, the graph $K_r \odot K_1$ and its complement are said to be *headless spiders*.

Proposition 4.4 ([AIM08, Proposition 3.14]): $Z(K_r \odot K_1) = r - 1$ *for* $r \geq 2$.

Proposition 4.5: $Z(\overline{K_r \odot K_1}) = 2r - 4$ *for* $r \geq 4$.

Proof. Observe that every vertex of $\overline{K_r \odot K_1}$ has degree $2(r-1)$ or $r-1$. These vertices are called *hubs* and *legs*, respectively. Since every vertex is adjacent to $r-1$ hubs, any forcing set must contain at least $r-2$ hubs; otherwise, no force is possible. Similarly, the legs can only be forced by hubs, but each hub is adjacent to $r-1$ legs, so any forcing set must contain at least $r-2$ legs. Thus, $Z(\overline{K_r \odot K_1}) \geq 2r-4$.

For the upper bound, construct a Z-sequence $S = (\ell_1, \ell_2, h_1, h_2)$ of $G := \overline{K_r \odot K_1}$ by taking two legs ℓ_1 and ℓ_2. The leg ℓ_1 footprints all hubs except one, which is then footprinted by ℓ_2. Choose a hub h_1 whose non-neighbor x is neither ℓ_1 nor ℓ_2. Then S is a Z-sequence for any hub $h_2 \neq h_1$ because h_1 footprints all $r \geq 4$ legs except those in $\{\ell_1, \ell_2, x\}$ and h_2 footprints x. Thus, $Z(G) = |G| - \mathrm{mz}(G) \leq |G| - |S| = 2r - 4$. ∎

Remark 4.6: The requirement $r \geq 4$ cannot be dropped from Proposition 4.5 as the co-net graph $\overline{K_3 \odot K_1}$ has forcing number 3 and $\overline{K_2 \odot K_1} = P_4$ has forcing number 1.

The next proposition provides a lower bound on the forcing number of spiders and similar graphs. It is based on *splits* (not to be confused with split graphs), which are defined as follows. Consider a graph $G = (V, E)$. A *split* is a partition of V into non-empty parts A and B such that the edges between A and B induce a complete bipartite subgraph [CMR12], see Figure 4.2. In other words, there are subsets $A' \subseteq A$ and $B' \subseteq B$ such that

$$E \cap \{ab \mid a \in A,\ b \in B\} = \{ab \mid a \in A',\ b \in B'\}. \tag{1}$$

Splits play a central role in split decompositions (introduced by Cunningham [Cun82] under the name *join decomposition*); see [BLS99, § 12.3] for a short survey.

Proposition 4.7: *Let G be a graph. If G has a split $V(G) = A \cup B$, then*

$$Z(G) \geq Z(G[A]) + Z(G[B]) - 2$$

Proof. Take $A' \subseteq A$ and $B' \subseteq B$ such that (1) holds. Suppose that $a \in A$ forces $b \in B$. Then $a \in A'$ and $b \in B'$ because every edge between A and B has an endpoint in A' and in B'. Since $B' \subseteq N(a)$, the vertex b is the only white vertex of B' when it is forced by a. Thus, there can be at most one force from A to B (and vice versa). In particular,

$$Z(G) \geq (Z(G[A]) - 1) + (Z(G[B]) - 1) = Z(G[A]) + Z(G[B]) - 2.$$ ∎

Corollary 4.8: *If G is a thick spider with r legs and head H, then*

$$Z(G) \geq Z(\overline{K_r \odot K_1}) + Z(H) - 2$$

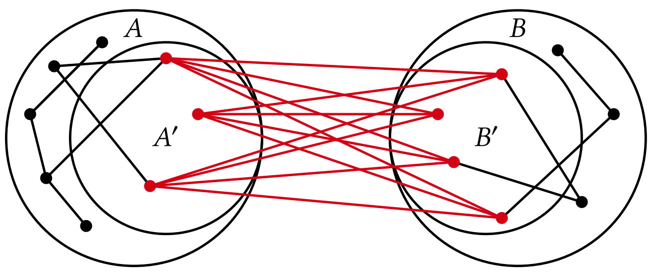


Figure 4.2: A graph with a split $V = A \cup B$. The red subgraph is complete bipartite.

Proposition 4.9: *Let G be a thick spider with $r \geq 2$ legs and head H. Then*

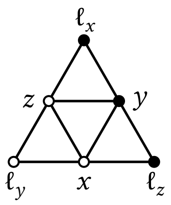
$$Z(G) = \begin{cases} r + Z(H) - \min\{2, n_0(H)\} & \text{if } r = 2, \\ r + Z(H) - \min\{1, n_0(H)\} & \text{if } r = 3, \\ 2r + Z(H) - 4 & \text{if } r \geq 4, \end{cases}$$

where $n_0(H)$ is the number of isolated vertices in H.

Proof. Let S_H be a minimum forcing set of H and let L be the set of legs in G.

Case 1 ($r = 2$): In this case, the statement follows from Proposition 4.3 since a thick spider with exactly two legs is also a thin spider with two legs.

Case 2 ($r = 3$): Let x, y, z be the vertices of K_r and let ℓ_x, ℓ_y, ℓ_z be their non-neighbors in G. We distinguish two subcases.



Case 2.1 (H is isolate-free): We claim that $S_G := \{\ell_z, y, \ell_x\} \cup S_H$ is a forcing set of G. The graph $G - H := \overline{K_3 \odot K_1}$ is depicted in the left margin with vertices in S_G colored black. The legs ℓ_z and ℓ_x can force x and z, respectively. Then, S_H can transitively force the rest of H and x can force ℓ_y. Thus, S_G is a forcing set of G and $Z(G) \leq |S_G| = r + Z(H)$.

By Corollary 3.4 and Proposition 4.2 we have $r + Z(H) = Z(K_r \vee H) \leq Z(G)$. This concludes the proof of $Z(G) = r + Z(H)$ for $n_0(H) = 0$.

Case 2.2 (H has an isolated vertex v): We claim that $S_G := \{\ell_z, y, \ell_x\} \cup S_H \setminus \{v\}$ is a forcing set of G. Color the vertices of S_G black. After x and z have been forced by ℓ_z and ℓ_x, the vertices of S_H can turn the rest of H black except for v. Then y can force v, after which x can force ℓ_y. Thus, S_G is a forcing set of G and $Z(G) \leq |S_G| = r + Z(H) - 1$.

By Corollary 4.8 and Remark 4.6, we have $Z(G) \geq r + Z(H) - 2$. This bound can only be attained if a hub forces a vertex of H and vice versa. Since K_r is a clique, its vertices must be all black before they can force a vertex of H. Thus, a force from H to K_r cannot occur after the opposite force. Now, assume that G has a forcing set S_G of size $r + Z(H) - 2$. Then S_G contains exactly $Z(H) - 1$ vertices from H and exactly $r - 1 = 2$ vertices from $G - H$. Suppose that the hub x is forced by some vertex of H. When this force occurs, the other hubs y and z must be black and H has a white vertex. This means that S_G contains at most one leg and that the other two legs are still white. Hence, every hub is adjacent to some white leg and some white vertex of H, so no force from K_r to H is possible, a contradiction. Therefore, $Z(G) \geq r + Z(H) - 1$.

In both subcases, we have shown that $Z(G) = r + Z(H) - \min\{1, n_0(H)\}$.

Case 3 ($r \geq 4$): In this case, we need to show $Z(G) = Z(H) + Z(G - H)$, where $G - H = \overline{K_r \odot K_1}$ is the graph induced by the vertices outside H. By Proposition 4.5, $Z(G - H) = 2r - 4$ and $G - H$ has a Z-sequence S consisting of two legs and two hubs. We can construct a Z-sequence of G by taking the two legs of S, appending a maximum Z-sequence of H, and then adding the two hubs of S. Thus, $Z(G) = |G| - \mathrm{mz}(G) \leq (2r + |H|) - (4 + \mathrm{mz}(H)) = 2r + Z(H) - 4$.

It remains to show that $Z(G) \geq 2r + Z(H) - 4$. Consider a minimum forcing set S of G. Since every vertex of G is adjacent to at least $r - 1$ hubs, S contains at least $r - 2$ hubs (otherwise no vertex of G can force). Similarly, S needs to contain at least $r - 2$ legs. Thus, $S \cap V(G - H) \geq 2r - 4 = Z(G - H)$ and a force from H to K_r would be useless since we can choose $S \cap V(G-H)$ to be a forcing set of $G-H$. The opposite force from a hub x to a vertex h of H requires all $r-1$ legs adjacent to x to be black. This means that S contains at least $r-1$ legs or that some leg was forced by a hub y, but y can only force a leg if h is black. In the latter case the force $x \to h$ cannot occur. In either case, we have $|S| \geq 2r + Z(H) - 4$. ∎

Theorem 4.10 ([JO92, Theorem 2.13]): *A graph G is P_4-sparse if and only if for every induced subgraph $H \neq K_1$ of G exactly one of the following statements holds:*

- *H is disconnected.*
- *$\overline{H}$ is disconnected.*
- *H is a spider or a headless spider.*

Since the class of P_4-sparse graphs is hereditary, we obtain with little work:

Corollary 4.11 ([JO92, Observation 2.12]): *A spider with head H is P_4-sparse if and only if H is P_4-sparse.*

Suppose that H is a graph whose complement $\overline{H}$ has the connected components $\overline{H_1}, \dots, \overline{H_k}$. Then H can be written as $H = \overline{\overline{H_1} \sqcup \dots \sqcup \overline{H_k}} = \bigvee_{i=1}^{k} H_i$ with union ($\sqcup$) and join ($\vee$) operations. Thus, the forcing number $Z(G)$ of a P_4-sparse graph G satisfies

$$Z(G) = \begin{cases} \sum_{i=1}^{k} Z(G_i) & \text{if } G \text{ has components } G_1, \dots, G_k, \\ |G| - \max_{i=1}^{k} \{\mathrm{mz}(G_i) + \min\{2, n_0(G_i)\}\} & \text{if } \overline{G} \text{ has components } \overline{G_1}, \dots, \overline{G_k}, \\ r + Z(H) - \min\{2, n_0(H)\} & \text{if } G \text{ is a thin spider with } r \text{ legs}, \\ r - 1 & \text{if } G \text{ is the corona } K_r \odot K_1 \ (r \geq 2), \\ 3 & \text{if } G \text{ is the co-net graph } \overline{K_3 \odot K_1}, \\ 2r - 4 & \text{if } G \text{ is the headless spider } \overline{K_r \odot K_1} \ (r \geq 4), \\ r + Z(H) - \min\{1, n_0(H)\} & \text{if } G \text{ is a thick spider with exactly 3 legs}, \\ 2r + Z(H) - 4 & \text{if } G \text{ is a thick spider with } r \geq 4 \text{ legs}, \end{cases}$$

where $\mathrm{mz}(G) = |G| - Z(G)$ is the Z-Grundy domination number, $n_0(G)$ is the number of isolated vertices in G, and H is G's head if G is a spider. The distinct degree sequences of thin, thick, and headless spiders enable us to efficiently distinguish the cases in the recursive formula above. Thus, the forcing number of a P_4-sparse graph can be computed using a linear-time dynamic programming algorithm. Its correctness follows from Observation 2.1, Theorem 3.2, Proposition 4.3 and 4.4, Remark 4.6, Proposition 4.5 and 4.9.

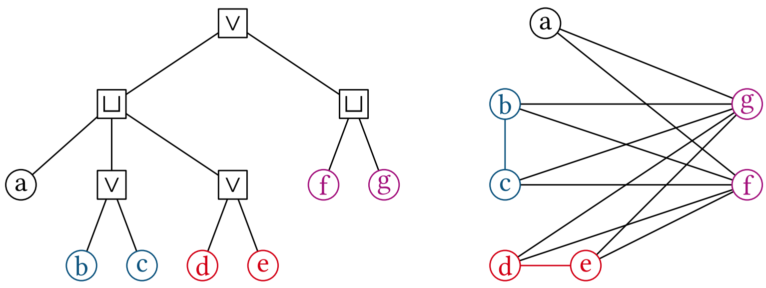


Figure 4.3: A $\{K_1\}$-cotree and the cograph it generates

4.1 Tree-Cographs

Let $\mathcal{F}$ be a family of graphs. An *$\mathcal{F}$-cotree* is a tree T where every leaf v represents a graph $G_v \in \mathcal{F}$ and every non-leaf is labeled $\vee$ (join) or $\sqcup$ (disjoint graph union). Given a vertex $v \in V(T)$, the subtree of T induced by v and its descendants is denoted T_v. The graph $\langle T_v \rangle$ *generated by* an $\mathcal{F}$-cotree T rooted at v is defined recursively

$$\langle T_v \rangle = \begin{cases} \bigvee_{d \in N(v)} \langle T_d \rangle & \text{if } v \text{ is labeled } \vee, \\ \bigsqcup_{d \in N(v)} \langle T_d \rangle & \text{if } v \text{ is labeled } \sqcup, \\ G_v & \text{if } v \text{ is a leaf of } T, \end{cases}$$

where $d \in N(v)$ if and only if d is a direct descendant (i.e. a child) of v. In [Kos19], the tree T_v is referred to as the modular decomposition tree of $\langle T_v \rangle$. It can be computed in linear time by decomposing $\langle T_v \rangle$ recursively [Kos19, Algorithm 5] and allows us to compute the forcing numbers Z, Z_t, and Z_c of $\langle T_v \rangle$ in linear time if we can compute them in linear time for graphs in $\mathcal{F}$.

Let $\mathrm{co}(\mathcal{F})$ denote the family of graphs generated by $\mathcal{F}$-cotrees. It is well-known that $\mathrm{co}(\{K_1\})$ is the family of cographs. In general, $\mathrm{co}(\mathcal{F})$ is the smallest superfamily of $\mathcal{F}$ that is closed under disjoint union and complementation. The class $\mathrm{co}(\mathcal{T})$, where $\mathcal{T}$ is the family of all trees, was studied by Tinhofer under the name *tree-cographs* [Tin89].

In the following, we determine the forcing numbers $Z(\overline{T})$ and $Z_c(\overline{T})$ of tree complements.

Proposition 4.12 ([Cur+24, Proposition 2.3])**:** *If T is a tree on $n \geq 4$ vertices, then*

$$Z(\overline{T}) = \begin{cases} n-1 & \text{if } T = K_{1,n-1}, \\ n-3 & \text{otherwise.} \end{cases}$$

The next result is well-known and easy to prove.

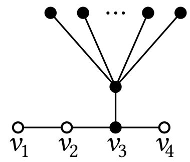
Proposition 4.13: *If T is a tree, then its complement $\overline{T}$ is disconnected if and only if $T = K_{1,n-1}$.*

Proposition 4.14: *If T is a tree on $n \geq 5$ vertices whose complement $\overline{T}$ is connected, then*

$$Z(\overline{T}) = Z_t(\overline{T}) = Z_c(\overline{T}) = n - 3.$$

Proof. Since $n \geq 5$, $\overline{T}$ is not a path. Thus, $Z(\overline{T}) \leq Z_t(\overline{T}) \leq Z_c(\overline{T})$. As the complement of the star $K_{1,n-1}$ is disconnected, $T \neq K_{1,n-1}$. By Proposition 4.12, we have $Z(\overline{T}) = n - 3$. It remains to show $Z_c(\overline{T}) \leq n - 3$ by constructing a connected forcing set of size $n - 3$.

As T is not a star, it contains an induced path on four vertices v_1, v_2, v_3, v_4 whose left endpoint v_1 is a leaf. Among all such paths, pick one path P where the degree of v_2 is maximal. In the complement $\overline{T}$, the vertices of P induce the path $v_3v_1v_4v_2$. Clearly, $S := V(T) \setminus \{v_1, v_2, v_4\}$ is a forcing set of $\overline{T}$. It remains to show that $T[S]$ is not a star, from which it follows by Proposition 4.13 that $\overline{T}[S]$ is connected.



Towards a contradiction, assume that $T[S]$ is a star. Then T is the graph shown on the right with vertices of S colored black. Consider the path $P^{-1} = v_4v_3v_2v_1$. Its second vertex v_3 has degree 3, while v_2 only has degree 2. This contradicts the choice of P. ∎

The propositions above show that the forcing numbers $Z(T)$, $Z_t(T)$, and $Z_c(T)$ can be computed in linear time for any tree T and its complement $\overline{T}$; hence, the same holds true for any tree-cograph.

Theorem 4.15: *The forcing numbers Z, Z_t, and Z_c can be computed in linear time on tree-cographs.*

5 Polynomial Kernels

In this chapter, we show that Connected Forcing does not admit a polynomial kernel unless coNP $\subseteq$ NP/poly.[1] It is believed that coNP $\nsubseteq$ NP/poly since the polynomial hierarchy would otherwise collapse to its third level [Jan13, § 2.1.3].

The notion of *kernelization* allows us to formally study pre-processing techniques for NP-hard problems. Intuitively speaking, a *kernelization algorithm* (or *kernel* for short) removes the easy parts of a parameterized problem to reveal its hard core. We use a slightly broader definition, where the core is not necessarily smaller than the original instance.

Definition 5.1 ([FLSZ19])**:** Let $L \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem over an alphabet Σ. A *kernel* for L is an algorithm with the following property: For any given $(G, d) \in \Sigma^* \times \mathbb{N}$, it outputs in time polynomial in $|G| + d$ a string $G' \in \Sigma^*$ and an integer $d' \in \mathbb{N}$ such that

$$[(G, d) \in L \iff (G', d') \in L] \quad \text{and} \quad |G'|, d' \leq f(d)$$

for some computable function f, which we call the *size* of the kernel. If f is polynomial, we say that the kernel is *polynomial*.

In this thesis, G is usually a graph, and we can set $|G| := |V(G)|$ instead of working with strings over Σ. The parameter d is often an integer between 0 and $|G|$. In this case, being polynomial in $|G| + d$ is equivalent to being polynomial in $|G|$.

It is well-known that a kernel for L exists if and only if L is fixed-parameter tractable (FPT). Kernels obtained from an FPT algorithm are often exponential in the parameter d, though.

The problems Zero Forcing, Total Forcing, and Connected Forcing can be solved in time $2^{\mathcal{O}(w^2)} \cdot |G|$, where w is the treewidth of G [Sch26]. Since $w \leq Z(G) \leq Z_t(G) \leq Z_c(G)$ for any connected graph G that is not a path, all three problems are FPT when parameterized by the solution size, but it is still open whether Zero Forcing admits a polynomial kernel [Sch26].

A common way to show that a parameterized problem does not admit a polynomial kernel is to give a *polynomial parameter transformation* (PPT) from a problem that does not have a polynomial kernel. PPTs are polynomial-time fpt-reductions with polynomial parameter growth. A formal definition is given below.

Definition 5.2 ([FLSZ19])**:** Let Π and Π' be two parameterized problems. A *PPT* from Π to Π' is an algorithm with the following property: *For* any given instance (G, d) of Π, it outputs in time polynomial in $|G| + d$ an instance (G', d') of Π' such that

$$[(G, d) \in \Pi \iff (G', d') \in \Pi'] \text{ and } d' \leq p(d)$$

[1] coNP is the class of decision problems for which the complement lies in NP, and NP/poly contains those problems a nondeterministic Turing machine can solve in polynomial time using an advice string that depends solely on the input size n and whose length is polynomial in n [Jan13, § 2.1.3].

for some polynomial p. This definition closely resembles that of polynomial kernels, but $|G'|$ may be unbounded in d. The next lemma establishes the usefulness of PPTs. We need one more definition to formulate it.

Definition 5.3 ([BTY11]): Let $\Pi \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem and let $\mathbf{1} \notin \Sigma$ be a "fresh" symbol. The derived *classical* (or *unparameterized*) problem Π_c of Π is the language $\left\{w\mathbf{1}^d \mid (w, d) \in \Pi\right\}$, that is, the parameter d is appended in unary onto the input word w.

Lemma 5.4 ([BTY11, Theorem 8]): *Let Π and Π' be parameterized problems whose derived classical problems Π_c, Π'_c are in NP. Suppose that Π_c is NP-complete, Π does not admit a polynomial kernel, and that there exists a PPT from Π to Π'. Then, Π' does not admit a polynomial kernel.*

A good candidate for Π is Red-Blue Dominating Set (RBDS), defined as follows.

Red-Blue Dominating Set (RBDS)	
Input:	A bipartite graph G, a 2-coloring $V(G) = R \cup B$ of its vertices, and an integer d
Parameter:	$d + \lvert B \rvert$
Question:	Is there a subset $R' \subseteq R$ of size at most d such that every vertex of B is adjacent to a vertex in R', that is, R' dominates B? In this case, we say that R' is a *red-blue dominating set* of G.

Unparameterized RBDS is known to be NP-complete. When parameterized by $d + |B|$, it does not admit a polynomial kernel unless coNP $\subseteq$ NP/poly [DLS14]. The next two observations on mandatory vertices in forcing sets are essential for our reduction of RBDS to Connected Forcing.

Observation 5.5 ([Fas17, Theorem 5.2]): *Let $G = (V, E)$ be a graph and F be non-empty. We say that F is a fort if no vertex of $V \setminus F$ has exactly one neighbor in F. In this case, any forcing set of G contains a vertex of F.*

Observation 5.6 ([DH19, Lemma 9]): *Let G be a graph. If v is a vertex with at least two leaf neighbors, then v is contained in every total forcing set (and in every connected forcing set).*

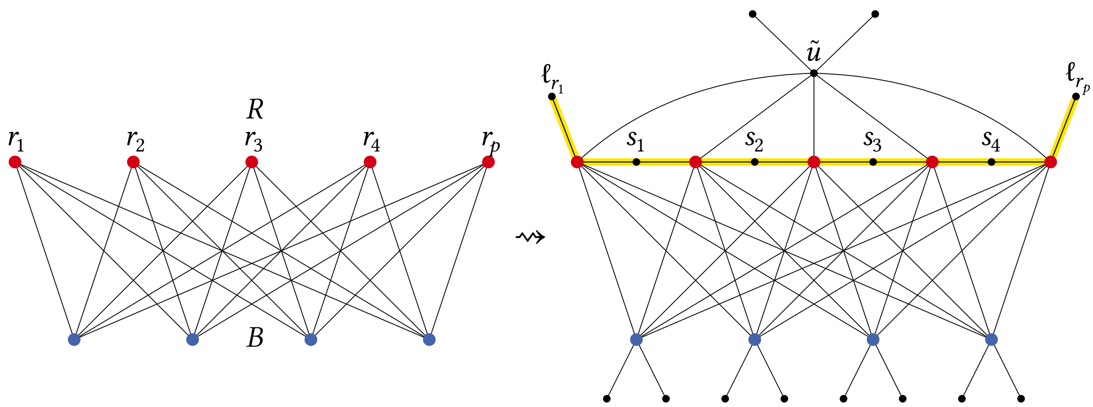


Figure 5.1: PPT from RBDS to Connected Forcing. Vertices of R and B are drawn red and blue, respectively. The induced path P is highlighted yellow.

Theorem 5.7: *Connected Forcing parameterized by solution size d does not have a polynomial kernel unless coNP $\subseteq$ NP/poly.*

Proof. It is clear that unparameterized Connected Forcing is in NP. By Lemma 5.4, it suffices to give a PPT from Red-Blue Dominating Set to Connected Forcing. Let (G, R, B, d) be an instance of RBDS and suppose that $G = (V, E) = (R \cup B, E)$. We construct an instance (G', d') of Connected Forcing as follows. First, we add a new vertex $\tilde{u}$ adjacent to all vertices of R. Next, we attach two leaves to $\tilde{u}$ and to each vertex in B. Then, we order the vertices of R linearly as $r_1, r_2, \dots, r_p$ and add a subdivided edge $r_i\, s_i\, r_{i+1}$ between consecutive vertices of R. Finally, we add two new vertices ℓ_{r_1} and ℓ_{r_p} adjacent only to r_1 and r_p, respectively. Notice that the vertices $\ell_{r_1} r_1\, s_1\, r_2\, s_2\, \dots\, r_p\, \ell_{r_p}$ form an induced path P of G' (see Figure 5.1). Formally, we define $G' = (V', E')$ where

$$V' = V \cup \{\tilde{u}, \ell^1_{\tilde{u}}, \ell^2_{\tilde{u}}\} \cup \{\ell^1_b, \ell^2_b \mid b \in B\} \cup \{s_i \mid 1 \le i < |R|\} \cup \{\ell_{r_1}, \ell_{r_p}\}, \text{ and}$$
$$E' = E \cup \{\tilde{u}r \mid r \in R\} \cup \{\tilde{u}\ell^1_{\tilde{u}}, \tilde{u}\ell^2_{\tilde{u}}\} \cup \{b\ell^1_b, b\ell^2_b \mid b \in B\} \cup \{r_i s_i, s_i r_{i+1} \mid 1 \le i < |R|\} \cup \{r_1\ell_{r_1}, r_p\ell_{r_p}\}.$$

We need to show that G' has a connected forcing set of size $d' := d + 2|B| + 3$ if and only if (G, R, B, d) is a *yes*-instance of RBDS. Suppose that G has a set $R' \subseteq R$ of size d that dominates B. We may assume that $d > 0$; otherwise, the RBDS problem is trivial. Choose an edge xy of P such that $x \in R'$. We claim that the set $S = R' \cup B \cup \{\ell^1_b \mid b \in B\} \cup \{\tilde{u}, \ell^1_{\tilde{u}}, y\}$ of size d' is a connected forcing set of G'. It is easy to see that $G'[S]$ is connected. Since x and y are adjacent vertices of the induced path P, they form a forcing set of P by Proposition 2.2. Thus, every white vertex of P can be forced black. The only remaining white vertices in G' are the leaves $\{\ell^2_b \mid b \in B\} \cup \{\ell^2_{\tilde{u}}\}$, each of which can be forced by its unique neighbor.

Conversely, suppose that G' has a connected forcing set S of size $d' = d + 2|B| + 3$. Then, by Observation 3.6, every vertex of $B \cup \{\tilde{u}\}$ has a leaf neighbor in S. Moreover, by Observation 5.6, we have $B \cup \{\tilde{u}\} \subseteq S$. Finally, S contains a vertex of $P \setminus R$ since $P \setminus R$ is a fort. In sum, we found $2|B| + 3$ vertices in $S \setminus R$, so $|S \cap R| \le d$. It remains to show that $S \cap R$ is a red-blue dominating set of G. Assume not, i.e. there is a vertex $b \in B$ that has no neighbor in $S \cap R$. But then b and $\tilde{u}$ would be in different components of $G'[S]$, contradicting the connectivity of $G'[S]$. ∎

Since G' is bipartite (by construction) and RBDS is NP-complete, we obtain the following.

Corollary 5.8: *Connected Forcing is NP-complete on bipartite graphs.*

Notice that the connectivity of $G[S]$ is the key property of Connected Forcing that enables the reduction from RBDS. The additional requirement that S must be a forcing set was just an obstacle that we addressed by adding the induced path P. Without P, this reduction is essentially the same as the reduction to Connected Vertex Cover by Dom et al. [DLS14].

The problems Vertex Cover and Feedback Vertex Set have polynomial kernels, but their connected variants do not [Mis+12]. Moreover, Vertex Cover already does not admit a polynomial kernel if we impose that every component induced by the solution set has at least two vertices. This phenomenon has been dubbed the "curse of connectivity" [FFPS10]. Careful examination of the proof of Theorem 5.7 shows that the same PPT[2] can be used to reduce RBDS to the following variant of zero forcing.

[2] The vertex $\tilde{u}$ and its leaves can be omitted when reducing RBDS to 3-Total Forcing.

3-Total Forcing	
Input:	A connected graph G and an integer d
Parameter:	d
Question:	Is there a forcing set $S \subseteq V(G)$ of size at most d such that every component of $G[S]$ has at least three vertices?

Thus, we can state the following.

Proposition 5.9: *3-Total Forcing is NP-complete even when restricted to bipartite graphs. It does not admit a polynomial kernel when parameterized by d unless coNP $\subseteq$ NP/poly.*

Proof. The proof is identical to that of Theorem 5.7 except for the last paragraph. Among all 3-total forcing sets of G' with size at most d', pick a set S such that the number of blue vertices dominated by $S \cap R$ is maximal. As before, we have $|S \cap R| \leq d$. If $S \cap R$ dominates all blue vertices, then $S \cap R$ is a red-blue dominating set of G, and we are done. Otherwise, let $b \in B$ be a blue vertex that has no neighbor in $S \cap R$. Since $b \in S$ by Observation 5.6 and every component of $G'[S]$ has at least three vertices, both leaf neighbors of b must be in S. Obtain S' from S by replacing one leaf neighbor ℓ with a red neighbor $r \in R \cap N(b)$, that is, $S' = (S \setminus \{\ell\}) \cup \{r\}$. Then S' is a 3-total forcing set of G' with size $|S'| \leq |S| \leq d'$, and $S' \cap R$ dominates more blue vertices than $S \cap R$, a contradiction. ∎

We were not able to obtain similar results for forcing problems with even weaker connectivity requirements. This is because the proof heavily relies on the fact that for every vertex $b \in B$, any forcing set S contains a leaf neighbor of b; thus S must also contain b and a red neighbor $r \in R \cap N(b)$ to meet the connectivity requirement.

6 W[P]-Complete Zero Forcing Variants

Let $k \geq 2$ be an integer. In this chapter, we show that the problems k-FORCING and PRECOLORED ZERO FORCING (defined below) are W[P]-complete. This settles their parameterized complexity by placing them at the top of the famous W-hierarchy

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[\mathrm{P}],$$

where all inclusions are believed to be proper. Under this assumption, both problems are much harder than ZERO FORCING, which is fixed-parameter tractable.

We can define W[P] as the class of parameterized problems decidable by a nondeterministic Turing machine in $f(d)n^{\mathcal{O}(1)}$ steps, at most $h(d)\log(n)$ of which are nondeterministic, where f and h are computable functions, d is the parameter, and n is the input size [FG06].

It is straightforward that k-FORCING and PRECOLORED ZERO FORCING belong to W[P] since a nondeterministic Turing machine may guess d vertices, encoded as a binary string of length $d\log(n)$. Whether these vertices form a k-forcing set can be checked in polynomial time.

To establish W[P]-hardness, we reduce from the following problem known to be W[P]-complete [ADF95, Corollary 3.6].

WEIGHTED MONOTONE CIRCUIT SATISFIABILITY (WMCS)	
Input:	A monotone Boolean circuit C and an integer d
Parameter:	d
Question:	Can C be satisfied by setting at most d sources to *true*?

Here, a monotone Boolean circuit C is a directed graph with n sources and a single sink. Every non-source of C is either an AND gate or an OR gate (the absence of NOT gates makes the circuit monotone). An assignment of truth values to the sources is said to *satisfy* C if the sink evaluates to *true* under the straightforward evaluation method. Setting all sources to *true* satisfies C, so it is natural to ask for the minimum number of sources needed to satisfy C.

As an intermediate step for proving W[P]-hardness, we introduce an auxiliary problem on directed graphs called AUX FORCING. In this problem, there are three types of vertices. A vertex labeled with OR (resp. AND) is required to have a single out-neighbor O and exactly two in-neighbors A and B. When it turns black, it forces O to become black provided that A or B (resp. A and B) are also black. Every remaining vertex is a *fork node*. When a fork node turns black, it turns every out-neighbor black. The problem AUX FORCING asks whether it is possible to color d vertices black such that the forcing process turns all other vertices black, where d is the parameter. In that case, the initial set of d black vertices is an *aux-forcing set*.

Lemma 6.1: *Aux Forcing is W[P]-hard.*

Proof. Let (C, d) be an instance of Weighted Monotone Circuit Satisfiability (WMCS). It suffices to give a polynomial-time transformation from the circuit C to a directed graph D such that D has an aux-forcing set of size d if and only if C can be satisfied by setting at most d sources to *true*. Importantly, the parameter d is preserved, so this is indeed an fpt-reduction.

We may assume that every logic gate of C has exactly two inputs, since for every $q \geq 3$, a q-ary logic gate can be replaced with an equivalent chain of $q - 1$ binary gates in polynomial time.

We build the directed graph D from C as follows. First, delete all n sources and make n disjoint copies of the remaining graph. Denote the i-th copy by C_i. For every non-sink vertex g in each copy, create a fork node f, rewire all outgoing edges of g to start at f instead, and insert the edge (g, f). Now, each logic gate has exactly one out-neighbor, as required for Aux Forcing. For each $i \in \{1, \dots, n\}$, create a fork node v_i corresponding to the source s_i of C. For every out-neighbor x of s_i, add the edge (v_i, x_i), where x_i is the copy of x in C_i. Finally, add an edge from the sink of C_i to v_i. See Figure 6.1 for an example of the resulting bipartite graph D.

Suppose that C has a satisfying assignment where only the sources $s_1, \dots, s_d$ are set to *true*. We claim that $Z := \{v_1, \dots, v_d\}$ is an aux-forcing set of D. Color the vertices in Z black. Then, for every gate in C that evaluates to *true*, the corresponding gate in every copy C_i can be forced black by construction. The vertex v_i, corresponding to the source s_i, has a single in-neighbor, which corresponds to the sink of C. As the sink of C evaluates to *true*, every vertex in $\{v_1, \dots, v_n\}$ can be turned black. Thus, Z can transitively force all of $\{v_1, \dots, v_n\}$ and hence all of $V(D)$.

Conversely, suppose that D has an aux-forcing set Z of size d. Let $z \in Z$. Unless $z \in \{v_1, \dots, v_n\}$, where n is the number of sources in C, there is some copy C_i, $1 \leq i \leq n$, of C that contains z. We claim that Z remains an aux-forcing set after swapping out z for v_i, that is, $Z' := (Z \setminus \{z\}) \cup \{v_i\}$ is an aux-forcing set of D. Any path from z to a vertex outside C_i must pass through v_i. Hence, the vertices outside C_i can still be forced after replacing z by v_i. In particular, $v_1, \dots, v_n$ can be forced, which in turn can transitively force all remaining vertices. This proves the claim. By swapping out every vertex in Z, we obtain an aux-forcing set of size at most d that only uses vertices from $\{v_1, \dots, v_n\}$. Thus, setting the corresponding sources in C to *true* satisfies C. ∎

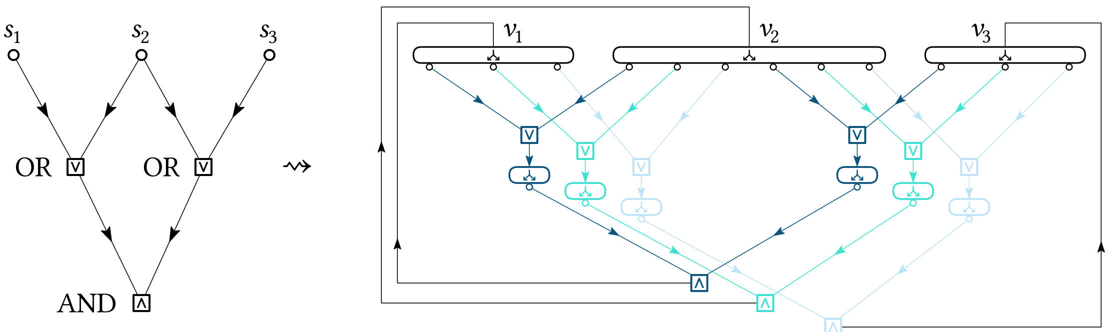


Figure 6.1: Reduction from WMCS to Aux Forcing, transforming circuit C into a directed graph D with OR gates ($\vee$), AND gates ($\wedge$), and fork nodes (⅄). Each copy C_i of the circuit C is drawn in a different shade of blue.

6.1 *k*-Forcing

Amos et al. [ACDP15] introduced k-forcing as a variant of zero forcing, where black vertices may force up to k neighbors instead of just one. A formal definition is given next. Let G be a graph (resp. a directed graph) and $k \in \mathbb{N}$ an integer. A set $S \subseteq V(G)$ of vertices is a *k-forcing set* if all vertices of G are black after the following iterated process:

- Initially, color all vertices of S black and the rest white.
- Whenever a black vertex has at most k white (out-)neighbors, it forces all of them to become black.

Note that 1-forcing is the same as (standard) zero forcing. The minimum size of a k-forcing set is called the *k-forcing number* $F_k(G)$. Since any k-forcing set is also a $(k+1)$-forcing set, we have $Z(G) = F_1(G) \geq F_2(G) \geq F_3(G)$ and so on. For $k \in \mathbb{N}$, the problem k-Forcing asks whether $F_k(G) \leq d$, where d is the parameter.

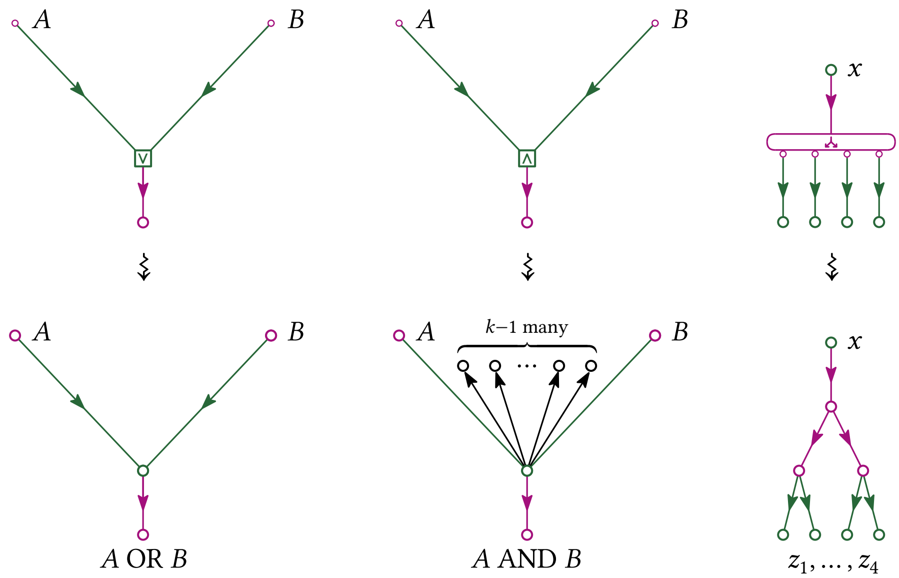


Figure 6.2: OR gadget, AND gadget, and fork gadget for k-forcing. The two undirected edges of the AND gadget are bidirectional. The colors green and purple (representing logic gates and fork nodes, respectively) show the bipartite structure of the Aux Forcing instance we are reducing from.

Lemma 6.2: *k-Forcing on directed graphs is W[P]-complete for every integer $k \geq 2$.*

Proof. Let (C, d) be an instance of WMCS where every logic gate has exactly two inputs. By Lemma 6.1, we obtain an equivalent Aux Forcing instance (D, d) in polynomial time. We now perform the local replacements sketched in Figure 6.2: OR gates are kept as is. Attach $k-1$ leaves to each AND gate and make its incoming edges bidirectional by adding an edge from it to every in-neighbor. Replace each fork node f with a binary out-tree on outdeg(f) leaves, and identify each out-neighbor of f with a unique leaf of the out-tree. See Figure A.1 for an example of the resulting graph D'.

Let Z be a minimum aux-forcing set of D and Z' be a minimum k-forcing set of D'. As before, we may assume that $Z, Z' \subseteq \{v_1, \dots, v_n\}$. Thus, the vertices A, B and x from Figure 6.2 are the only vertices of each gadget that may be black at the start of the forcing process; the others are white. By inspection, we see that the sink(s) of the OR gadget (AND gadget) can be forced if and only if A or B is black (A and B are black). Further, the leaves $z_1, \dots, z_\ell$ of the fork gadget only turn black when x becomes black. That is, the gadgets act according to the Aux Forcing rules. Therefore, $|Z'| = |Z|$ and we have $|Z'| \leq d$ if and only if (C, d) is a *yes*-instance of WMCS. ∎

Next, we extend this result to undirected graphs. We need to be careful, though, since replacing every directed edge with a one-way gadget from Figure 6.4 may not preserve the k-forcing number, as the following example shows: Consider the orientation of K_4 in Figure 6.3: any vertex forms a 2-forcing set of size 1, but after all edges are replaced with one-way gadgets, each vertex has degree at least 3; hence, it cannot force its neighbors.

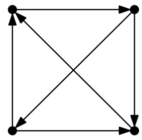


Figure 6.3

Theorem 6.3: *k-Forcing is W[P]-complete for every integer $k \geq 2$.*

Proof. Let (C, d) be an instance of WMCS where every logic gate has exactly two inputs. By Lemma 6.2, we obtain an equivalent k-Forcing instance (D, d) in polynomial time, where D is a directed graph. Transform D into an undirected graph G by replacing bidirectional edges with undirected edges and directed edges with one-way gadgets as follows: First, create an empty graph G with vertex set $V(D)$. Then, for every edge (s, t) of D such that (t, s) is not an edge, attach a star S with k leaves to s in G. Take a leaf of S and attach two leaves to it. Now, S has $k + 1$ leaves. Add a new vertex v adjacent to every leaf of S, then add the edge vt. This creates a one-way gadget as shown in Figure 6.4. Finally, replace every pair $((s, t), (t, s))$ of bidirectional edges with a single undirected edge st to keep the AND gadgets intact. To prove the theorem, it suffices to show $F_k(G) = F_k(D)$.

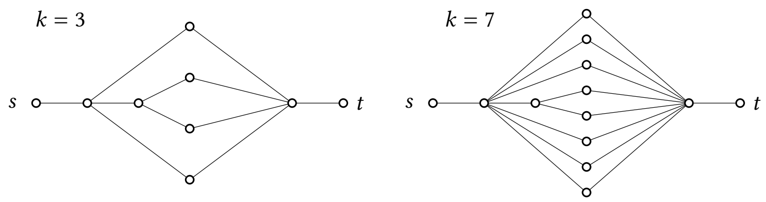


Figure 6.4: One-way gadget to simulate a directed edge (s, t). Note that $\{s\}$ is a k-forcing set of the gadget, but $\{t\}$ is not because the $k + 1$ central vertices cannot be forced from the right.

Let Z be a minimum k-forcing set of D. As usual, we may assume that $Z \subseteq \{v_1, \dots, v_n\}$, where v_i corresponds to the source s_i of C. Consider a forcing process starting from Z that turns $V(D)$ black. In this process, a black vertex x can force whenever it has at most k white out-neighbors, independent of the colors of its in-neighbors. However, if x does not correspond to an OR gate of C, all its in-neighbors must be black by construction of D. Thus, x can also force in the undirected graph G, even though in-neighbors are counted there. Further, if x does correspond to an OR gate, then x has exactly three neighbors in G, one of which is black. Hence, x has at most $2 \leq k$ white neighbors in G and can still force. Therefore, Z is also a k-forcing set of G. Consequently, $F_k(G) \leq |Z| = F_k(D)$.

Conversely, suppose that G has a k-forcing set Z of size strictly smaller than $F_k(D)$. Then we can replace every vertex $z \in (Z \cap V(G)) \setminus V(D)$ with the left endpoint s of the one-way gadget containing z. This yields a k-forcing set for the directed graph D of size strictly smaller than $F_k(D)$, a contradiction. ∎

6.2 Precolored Zero Forcing

In this section, we show that the following problem is W[P]-complete.

Precolored Zero Forcing	
Input:	A graph G, a set $B \subseteq V(G)$ of precolored vertices and an integer d
Parameter:	d
Question:	Is there a set $S \subseteq V(G)$ of size at most d such that $S \cup B$ is a forcing set?

Essentially, the problem is to determine the minimum number of vertices that need to be added to a set of precolored vertices in order to obtain a forcing set. Previously, it was only known that Precolored Zero Forcing is W[2]-hard [Caz+19, Theorem 2].

Lemma 6.4: *Precolored Zero Forcing on directed graphs is W[P]-complete.*

Proof. Let (C, d) be an instance of WMCS where every logic gate has exactly two inputs. By Lemma 6.1, we obtain an equivalent Aux Forcing instance (D, d) in polynomial time. We now perform the local replacements sketched in Figure 6.5: For each OR gate g, subdivide its outgoing edge, precolor the subdivision vertex v, and add the backwards edge (v, g). For each AND gate g, subdivide its incoming edges, then add an edge from g to each subdivision vertex. For each fork node f, subdivide its outgoing edges twice, then precolor every out-neighbor y_i of f in the resulting graph and add the backwards edge (y_i, f) for $1 \leq i \leq \text{outdeg}(f)$. See Figure A.2 for an example of the resulting graph D'. The remainder of the proof is analogous to that of Lemma 6.2. ∎

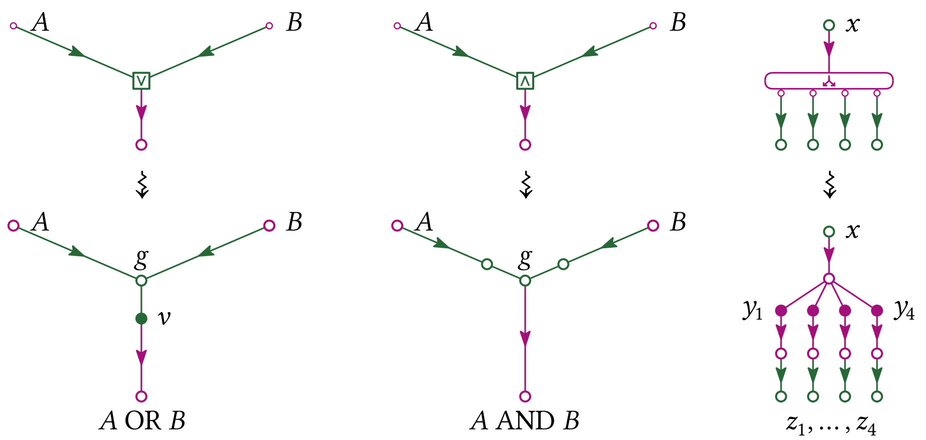


Figure 6.5: OR gadget, AND gadget, and fork gadget for Precolored Zero Forcing, with bidirectional edges drawn as undirected edges and precolored vertices as filled vertices.

Theorem 6.5: *Precolored Zero Forcing is W[P]-complete.*

Proof. Let (C, d) be an instance of WMCS where every logic gate has exactly two inputs. By Lemma 6.4, we obtain an equivalent Precolored Zero Forcing instance (D, d) in polynomial time, where D is a directed graph with precolored vertices. Transform D into an undirected graph G by replacing bidirectional edges with undirected edges and every directed edge with the one-way gadget of Figure 6.6.

Let Z be a minimum set of vertices such that $Z \cup P$ is a forcing set of D, where P is the set of precolored vertices. By swapping out vertices of Z, we may assume that $Z \subseteq \{v_1, \dots, v_n\}$, where v_i corresponds to the source s_i of C. Consider a sequence of forces starting from $Z \cup P$ that turns $V(D)$ black. As in the proof of Theorem 6.3, we only need to show that for every force $x \to y$, there is an equivalent process in G that turns y black when x is black. Again, only the case where x corresponds to an OR gate is interesting. In this case, x has a precolored out-neighbor in G, which can force when x is black. Therefore, $Z \cup P$ is also a forcing set of G.

Conversely, suppose that G has a forcing set $Z' \cup P$ such that $|Z'| < |Z|$. Then we can replace every vertex $z \in Z' \cap V(G) \setminus V(D)$ with the left endpoint s of the one-way gadget containing z, contradicting the minimality of Z. ∎

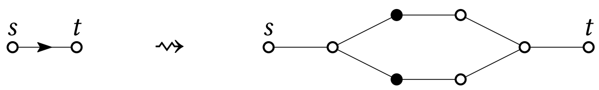


Figure 6.6: One-way gadget to simulate a directed edge (s, t). The filled vertices are precolored. Note that $\{s\}$ is a forcing set of the gadget, but $\{t\}$ is not.

7 Concluding Remarks

We have shown that Zero Forcing can be solved in linear time on P_4-sparse graphs. It is expected that this result can be generalized to P_4-tidy graphs with little difficulty. Even more generally, one can ask whether Zero Forcing is FPT when parameterized by clique-width. Note that cographs are exactly the graphs of clique-width at most 2 [CO00], and it is still open whether Zero Forcing can be solved in polynomial time on distance-hereditary graphs, which have clique-width at most 3. Besides, the classical complexity of Zero Forcing on split graphs and interval graphs is unknown.

It would also be interesting to know the parameterized complexity of Restricted Zero Forcing [Boz+19]. Given a set of required vertices, this problem asks for a minimum superset that is a zero forcing set. Its unparameterized version is equivalent to Precolored Zero Forcing. However, Restricted Zero Forcing has a bigger parameter, so it might be easier than Precolored Zero Forcing, which is W[P]-complete.

Perhaps the most important open question is whether Zero Forcing and Total Forcing admit a polynomial kernel. I suspect they do not, based on the negative results for Connected Forcing and 3-Total Forcing. In particular, such a kernel should be able to compress any Möbius ladder to an equivalent instance of constant size since every Möbius ladder has zero forcing number 4 [AIM08, Proposition 3.9]. Many other parametric graph families of constant zero forcing number exist [Hog+], but existing reduction rules from the Zero Forcing literature do not suffice to compress all of them in a generic manner.

Bibliography

[ACDP15] David Amos, Yair Caro, Randy Davila, and Ryan Pepper. “Upper bounds on the k-forcing number of a graph”. In: *Discrete Applied Mathematics* **181** (2015), pp. 1–10. DOI: 10.1016/j.dam.2014.08.029. MR: 3284506.

[ADF95] Karl A. Abrahamson, Rodney G. Downey, and Michael R. Fellows. “Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogues”. In: *Annals of Pure and Applied Logic* **73** (1995), pp. 235–276. DOI: 10.1016/0168-0072(94)00034-Z. MR: 1336643.

[AIM08] AIM Minimum Rank – Special Graphs Work Group. “Zero forcing sets and the minimum rank of graphs”. In: *Linear Algebra and its Applications* **428** (2008), pp. 1628–1648. DOI: 10.1016/j.laa.2007.10.009. MR: 2388646.

[AM25] Aida Abiad and Maryam Moghaddas. “On the Z_q-forcing number: computational approach and exact values”. In: *Journal of Physics: Complexity* **6**, 045005 (2025). DOI: 10.1088/2632-072X/ae1430.

[Bar+10] Francesco Barioli, Wayne Barrett, Shaun M. Fallat, H. Tracy Hall, Leslie Hogben, Bryan Shader, P. van den Driessche, and Hein van der Holst. “Zero forcing parameters and minimum rank problems”. In: *Linear Algebra and its Applications* **433** (2010), pp. 401–411. DOI: 10.1016/j.laa.2010.03.008. MR: 2645093.

[Bar+13] Francesco Barioli, Wayne Barrett, Shaun M. Fallat, H. Tracy Hall, Leslie Hogben, Bryan Shader, P. van den Driessche, and Hein van der Holst. “Parameters Related to Tree-Width, Zero Forcing, and Maximum Nullity of a Graph”. In: *Journal of Graph Theory* **72** (2013), pp. 146–177. DOI: 10.1002/jgt.21637. MR: 3010007.

[BDS25] Boris Brimkov, Randy Davila, and Houston Schuerger. “Connected forcing density and related problems”. 2025. arXiv: 2507.11194.

[BG07] Daniel Burgarth and Vittorio Giovannetti. “Full Control by Locally Induced Relaxation”. In: *Physical Review Letters* **99**, 100501 (2007). DOI: 10.1103/PhysRevLett.99.100501.

[BG25] Thomas Bläsius and Max Göttlicher. “An Efficient Algorithm for Power Dominating Set”. In: *Algorithmica* **87** (2025), pp. 344–376. DOI: 10.1007/s00453-024-01283-8. MR: 4867972.

[BHHS25] Wayne Barrett, Mark Hunnell, John Hutchens, and John Sinkovic. “The classification of graphs on eight vertices with coinciding zero forcing number and maximum nullity”. In: *Electronic Journal of Linear Algebra* **41** (2025), pp. 643–668. DOI: 10.13001/ela.2025.9635. MR: 5024562.

[BLS99] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. SIAM, 1999. ISBN: 0-89871-432-X. DOI: 10.1137/1.9780898719796. MR: 1686154.

[BO95] Luitpold Babel and Stephan Olariu. “On the isomorphism of graphs with few P_4's”. In: *Graph-theoretic concepts in computer science (Aachen, 1995)*. Springer, 1995, pp. 24–36. ISBN: 3-540-60618-1. DOI: 10.1007/3-540-60618-1_63. MR: 1400010.

[Bod98] Hans L. Bodlaender. “A partial k-arboretum of graphs with bounded treewidth”. In: *Theoretical Computer Science* **209** (1998), pp. 1–45. DOI: 10.1016/S0304-3975(97)00228-4. MR: 1647486.

[Boz+19] Chassidy Bozeman, Boris Brimkov, Craig Erickson, Daniela Ferrero, Mary Flagg, and Leslie Hogben. “Restricted power domination and zero forcing problems”. In: *Journal of Combinatorial Optimization* **37** (2019), pp. 935–956. DOI: 10.1007/s10878-018-0330-6. MR: 3928418.

[Bre+17] Boštjan Brešar, Csilla Bujtás, Tanja Gologranc, Sandi Klavžar, Gašper Košmrlj, Balázs Patkós, Zsolt Tuza, and Máté Vizer. “Grundy dominating sequences and zero forcing sets”. In: *Discrete Optimization* **26** (2017), pp. 66–77. DOI: 10.1016/j.disopt.2017.07.001. MR: 3720282.

[BS19] Boris Brimkov and Zachary Scherr. “An exact algorithm for the minimum rank of a graph”. 2019. arXiv: 1912.00158.

[BTY11] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. “Kernel bounds for disjoint cycles and disjoint paths”. In: *Theoretical Computer Science* **412** (2011), pp. 4570–4578. DOI: 10.1016/j.tcs.2011.04.039. MR: 2850415.

[Cam+26] Thomas R. Cameron, Leslie Hogben, Franklin H. J. Kenter, Seyed Ahmad Mojallal, and Houston Schuerger. “Forts, (fractional) zero forcing, and Cartesian products of graphs”. 2026. arXiv: 2310.17904v3.

[Caz+19] Pierre Cazals, Benoit Darties, Annie Chateau, Rodolphe Giroudeau, and Mathias Weller. “Power edge set and zero forcing set remain difficult in cubic graphs”. In: *Combinatorial algorithms*. Springer, 2019, pp. 122–135. ISBN: 978-3-030-25004-1. DOI: 10.1007/978-3-030-25005-8_11. MR: 3991230.

[CLTC26] Chaohui Chen, Muhuo Liu, Bit-Shun Tam, and Bo Cheng. “Bounds for zero forcing numbers of connected graphs with fixed order and maximum degree”. In: *Discrete Applied Mathematics* **379** (2026), pp. 116–139. DOI: 10.1016/j.dam.2025.08.001. MR: 4944011.

[CMR12] Pierre Charbit, Fabien de Montgolfier, and Mathieu Raffinot. “Linear time split decomposition revisited”. In: *SIAM Journal on Discrete Mathematics* **26** (2012), pp. 499–514. DOI: 10.1137/10080052X. MR: 2967479.

[CO00] Bruno Courcelle and Stephan Olariu. “Upper bounds to the clique width of graphs”. In: *Discrete Applied Mathematics* **101** (2000), pp. 77–114. DOI: 10.1016/S0166-218X(99)00184-5. MR: 1743732.

[Col90] Yves Colin de Verdière. “Sur un nouvel invariant des graphes et un critère de planarité”. In: *Journal of Combinatorial Theory. Series B* **50** (1990), pp. 11–21. DOI: 10.1016/0095-8956(90)90093-F. MR: 1070462.

[Cun82] William H. Cunningham. “Decomposition of directed graphs”. In: *Society for Industrial and Applied Mathematics. Journal on Algebraic and Discrete Methods* **3** (1982), pp. 214–228. DOI: 10.1137/0603021. MR: 655562.

[Cur+24] Emelie Curl, Shaun Fallat, Ryan Moruzzi Jr, Carolyn Reinhart, and Derek Young. "On the zero forcing number of the complement of graphs with forbidden subgraphs". In: *Linear Algebra and its Applications* **703** (2024), pp. 187–207. DOI: 10.1016/j.laa.2024.09.009. MR: 4797230.

[Cyg+15] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms.* Springer, 2015. ISBN: 978-3-319-21274-6. DOI: 10.1007/978-3-319-21275-3. MR: 3380745.

[DH19] Randy Davila and Michael A. Henning. "On the total forcing number of a graph". In: *Discrete Applied Mathematics* **257** (2019), pp. 115–127. DOI: 10.1016/j.dam.2018.09.001. MR: 3926378.

[DHP23] Randy Davila, Michael A. Henning, and Ryan Pepper. "Zero and total forcing dense graphs". In: *Discussiones Mathematicae Graph Theory* **43** (2023), pp. 619–634. DOI: 10.7151/dmgt.2389. MR: 4590478.

[Die25] Reinhard Diestel. *Graph theory.* Sixth edition. Springer, 2025. ISBN: 978-3-662-70106-5. DOI: 10.1007/978-3-662-70107-2. MR: 4874150.

[DLS14] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. "Kernelization Lower Bounds Through Colors and IDs". In: *ACM Transactions on Algorithms* **11**, Article 13 (2014). DOI: 10.1145/2650261. MR: 3283568.

[DY26] Louis Deaett and Derek Young. "Relationships between minimum rank problem parameters for cobipartite graphs". In: *Discrete Applied Mathematics* **381** (2026), pp. 378–397. DOI: 10.1016/j.dam.2025.12.022. MR: 5006242.

[EKY17] Linda Eroh, Cong X. Kang, and Eunjeong Yi. "A comparison between the metric dimension and zero forcing number of trees and unicyclic graphs". In: *Acta Mathematica Sinica, English Series* **33** (2017), pp. 731–747. DOI: 10.1007/s10114-017-4699-4. MR: 3648519.

[Fas17] Caleb C. Fast. "Novel Techniques for the Zero-Forcing And p-Median Graph Location Problems". Rice University, 2017. HDL: 1911/102254. MR: 3781779.

[Fer+25] Daniela Ferrero, H. Tracy Hall, Leslie Hogben, Mark Hunnell, and Ben Small. "Zero forcing propagation time intervals and graphs with fixed propagation time". 2025. arXiv: 2511.16335.

[FFPS10] Henning Fernau, Fedor V. Fomin, Geevarghese Philip, and Saket Saurabh. "The curse of connectivity: t-total vertex (edge) cover". In: *Computing and combinatorics.* Springer, 2010, pp. 34–43. ISBN: 978-3-642-14030-3. DOI: 10.1007/978-3-642-14031-0_6. MR: 2720079.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Springer, 2006. ISBN: 978-3-540-29952-3. DOI: 10.1007/3-540-29953-X. MR: 2238686.

[FH07] Shaun M. Fallat and Leslie Hogben. "The minimum rank of symmetric matrices described by a graph: A survey". In: *Linear Algebra and its Applications* **426** (2007), pp. 558–582. DOI: 10.1016/j.laa.2007.05.036. MR: 2350678.

[FLSZ19] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing.* Cambridge University Press, 2019. ISBN: 978-1-107-05776-0. MR: 3887686.

[FMY16] Shaun Fallat, Karen Meagher, and Boting Yang. “On the complexity of the positive semidefinite zero forcing number”. In: *Linear Algebra and its Applications* (2016), pp. 101–122. DOI: 10.1016/j.laa.2015.03.011. MR: 3440126.

[FR20] Maximilian Fürst and Dieter Rautenbach. “A short proof for a lower bound on the zero forcing number”. In: *Discussiones Mathematicae Graph Theory* **40** (2020), pp. 355–360. DOI: 10.7151/dmgt.2117. MR: 4041987.

[Hal26] H. Tracy Hall. “The Delta Theorem: a dimension bound for faithful orthogonal graph representations”. 2026. arXiv: 2601.01211.

[HCY10] Liang-Hao Huang, Gerard J. Chang, and Hong-Gwa Yeh. “On minimum rank and zero forcing sets of a graph”. In: *Linear Algebra and its Applications* **432** (2010), pp. 2961–2973. DOI: 10.1016/j.laa.2010.01.001. MR: 2639259.

[HHHH02] Teresa W. Haynes, Sandra M. Hedetniemi, Stephen T. Hedetniemi, and Michael A. Henning. “Domination in Graphs Applied to Electric Power Networks”. In: *SIAM Journal on Discrete Mathematics* **15** (2002), pp. 519–529. DOI: 10.1137/S0895480100375831. MR: 1935835.

[HHK21] Hongbo Hua, Xinying Hua, and Sandi Klavžar. “Zero forcing number versus general position number in tree-like graphs”. 2021. arXiv: 2112.09999.

[HL84] S. T. Hedetniemi and Renu Laskar. “Connected domination in graphs”. In: *Graph theory and combinatorics (Cambridge, 1983)*. Academic Press, London, 1984, pp. 209–217. ISBN: 0-12-111760-X. IA: graphtheorycombi0000camb_z9v1. MR: 777177.

[HLS22] Leslie Hogben, Jephian C.-H. Lin, and Bryan L. Shader. *Inverse problems and zero forcing for graphs*. American Mathematical Society, Providence, RI, 2022. ISBN: 978-1-4704-6655-8. DOI: 10.1090/surv/270. MR: 4478249.

[Hog+] Leslie Hogben, Wayne Barrett, Jason Grout, Hein van der Holst, Kaela Rasmussen, Andrew Smith, and Derek Young. “AIM minimum rank graph catalog”. URL: http://admin.aimath.org/resources/graph-invariants/minimumrankoffamilies/.

[Jan13] Bart M. P. Jansen. “The Power of Data Reduction: Kernels for Fundamental Graph Problems”. Utrecht University, 2013. ISBN: 978-90-393-5966-2. HDL: 1874/276438.

[JIBR16] I. Javaid, I. Irshad, M. Batool, and Z. Raza. “On the zero forcing number of corona and lexicographic product of graphs”. 2016. arXiv: 1607.04071.

[JO92] B. Jamison and S. Olariu. “A tree representation for P_4-sparse graphs”. In: *Discrete Applied Mathematics* **35** (1992), pp. 115–129. DOI: 10.1016/0166-218X(92)90036-A. MR: 1137806.

[JZJ23] Yu Jing, Wenqian Zhang, and Shengjin Ji. “The zero forcing number of graphs with the matching number and the cyclomatic number”. In: *Graphs and Combinatorics* **39**, Article 72 (2023). DOI: 10.1007/s00373-023-02664-6. MR: 4607718.

[Kos19] Tim Kos. “Contributions to the Study of Contemporary Domination Invariants of Graphs”. University of Maribor, 2019. HDL: 20.500.12556/DKUM-73378.

[KS96] Haim Kaplan and Ron Shamir. “Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques”. In: *SIAM Journal on Computing* **25** (1996), pp. 540–561. DOI: 10.1137/S0097539793258143. MR: 1390027.

[Mis+12] Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, Saket Saurabh, and Somnath Sikdar. “FPT algorithms for Connected Feedback Vertex Set”. In: *Journal of Combinatorial Optimization* **24** (2012), pp. 131–146. DOI: 10.1007/s10878-011-9394-2. MR: 2943658.

[Owe09] Kayla Denise Owens. “Properties of the Zero Forcing Number”. Brigham Young University, 2009. URL: https://scholarsarchive.byu.edu/etd/2216.

[Row11a] Darren D. Row. “Zero forcing number, path cover number, and maximum nullity of cacti”. In: *Involve* **4** (2011), pp. 277–291. DOI: 10.2140/involve.2011.4.277. MR: 2905229.

[Row11b] Darren Daniel Row. “Zero forcing number: Results for computation and comparison with other graph parameters”. Iowa State University, 2011. DOI: 10.31274/etd-180810-76. MR: 2941885.

[Sch26] Robert Scheffler. “On the Parameterized Complexity of Grundy Domination and Zero Forcing Problems”. 2026. arXiv: 2508.18104v3.

[Tak13] Fatemeh Alinaghipour Taklimi. “Zero Forcing Sets for Graphs”. University of Regina, 2013. HDL: 10294/5477. MR: 4272166.

[Tin89] G. Tinhofer. “Strong tree-cographs are birkhoff graphs”. In: *Discrete Applied Mathematics* **22** (1988/89), pp. 275–288. DOI: 10.1016/0166-218X(88)90100-X. MR: 0983251.

[TUK95] Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. “Mixed searching and proper-path-width”. In: *Theoretical Computer Science* **137** (1995), pp. 253–268. DOI: 10.1016/0304-3975(94)00160-K. MR: 1311224.

[WWZ20] Xinlei Wang, Dein Wong, and Yuanshuai Zhang. “Zero forcing number of a graph in terms of the number of pendant vertices”. In: *Linear and Multilinear Algebra* **68** (2020), pp. 1424–1433. DOI: 10.1080/03081087.2018.1545829. MR: 4137055.

[Yan13] Boting Yang. “Fast–mixed searching and related problems on graphs”. In: *Theoretical Computer Science* (2013), pp. 100–113. DOI: 10.1016/j.tcs.2013.04.015. MR: 3110517.

[ZWWJ22] Wenqian Zhang, Jianfeng Wang, Weifan Wang, and Shengjin Ji. “On the Zero Forcing Number and Spectral Radius of Graphs”. In: *The Electronic Journal of Combinatorics* **29**, P1.33 (2022). DOI: 10.37236/10638. MR: 4395936.

A Appendix

List of computational problems

Zero Forcing	
Input:	A graph G and an integer d
Parameter:	d
Question:	Is there a forcing set $S \subseteq V(G)$ of size at most d?

Total Forcing	
Input:	An isolate-free graph G and an integer d
Parameter:	d
Question:	Is there a forcing set $S \subseteq V(G)$ of size at most d such that S induces an isolate-free subgraph?

3-Total Forcing	
Input:	A connected graph G and an integer d
Parameter:	d
Question:	Is there a forcing set $S \subseteq V(G)$ of size at most d such that every component of $G[S]$ has at least three vertices?

Connected Forcing	
Input:	A connected graph G and an integer d
Parameter:	d
Question:	Is there a forcing set $S \subseteq V(G)$ of size at most d such that S induces a connected subgraph?

Dominating Set	
Input:	A graph $G = (V, E)$ and an integer d
Parameter:	d
Question:	Is there a subset $D \subseteq V$ of size at most d such that $N[D] = V$, i.e. every vertex $v \in V \setminus D$ has a neighbor in D?

Red-Blue Dominating Set	
Input:	A bipartite graph G, a 2-coloring $V(G) = R \cup B$ of its vertices, and an integer d
Parameter:	$d + \lvert B \rvert$
Question:	Is there a subset $R' \subseteq R$ of size at most d such that $N(R') = B$, i.e. every vertex of B is adjacent to a vertex in R'?

Connected Vertex Cover	
Input:	A graph $G = (V, E)$ and an integer d
Parameter:	d
Question:	Is there a subset $C \subseteq V$ of size at most d such that $G[C]$ is connected and every edge $uv \in E$ has an endpoint in C, i.e. $u \in C$ or $v \in C$?

Feedback Vertex Set	
Input:	A graph $G = (V, E)$ and an integer d
Parameter:	d
Question:	Is there a subset $F \subseteq V$ of size at most d such that $G[V \setminus F]$ is a forest?

Weighted Monotone Circuit Satisfiability	
Input:	A boolean circuit C without NOT gates and an integer d
Parameter:	d
Question:	Can C be satisfied by setting at most d sources to *true*?

k-Forcing	
Input:	A graph G and an integer d
Parameter:	d
Question:	Is there a k-forcing set of size at most d?

Precolored Zero Forcing	
Input:	A graph G, a set $B \subseteq V(G)$ of precolored vertices and an integer d
Parameter:	d
Question:	Is there a set $S \subseteq V(G)$ of size at most d such that $S \cup B$ is a forcing set?

Restricted Zero Forcing	
Input:	A graph G, a set $B \subseteq V(G)$ of required vertices and an integer d
Parameter:	d
Question:	Does G have a forcing set $S \supseteq B$ of size at most d?

Power Dominating Set	
Input:	A graph G and an integer d
Parameter:	d
Question:	Is there a set $S \subseteq V(G)$ of size at most d whose closed neighborhood $N[S]$ is a forcing set?

Figures

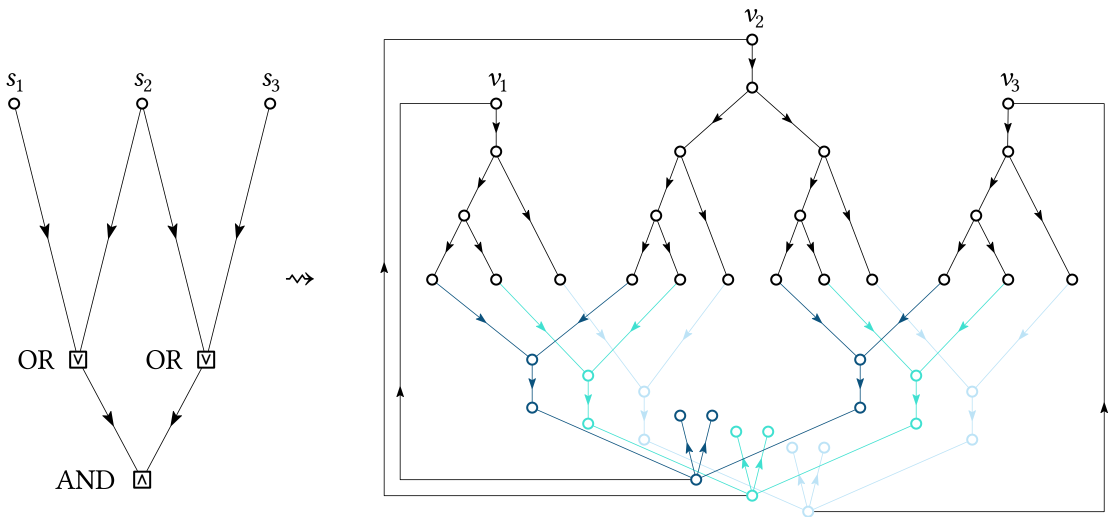


Figure A.1: Reduction from WMCS to 3-forcing on directed graphs, with bidirectional edges drawn as undirected edges

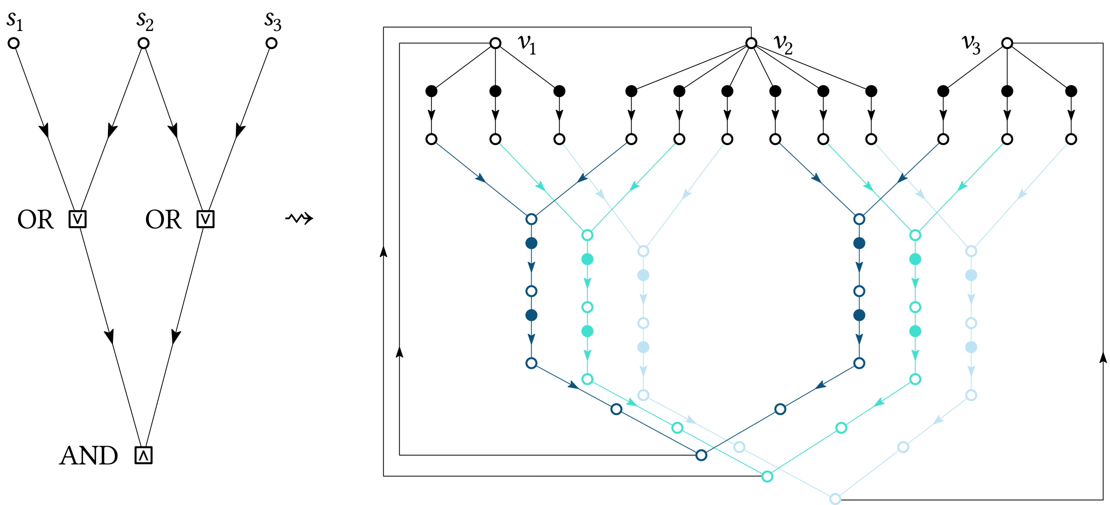


Figure A.2: Reduction from WMCS to Precolored Zero Forcing on directed graphs, with bidirectional edges drawn as undirected edges.