

EGERVÁRY RESEARCH GROUP
ON COMBINATORIAL OPTIMIZATION



TECHNICAL REPORTS

TR-2022-02. Published by the Egerváry Research Group, Pázmány P. sétány 1/C,
H-1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres. ISSN 1587-4451.

Resolving Infeasibility of Linear Systems: A Parameterized Approach

Kristóf Bérczi, Alexander Göke,
Lydia Mirabel Mendoza-Cadena, and
Matthias Mnich

January 2022

Resolving Infeasibility of Linear Systems: A Parameterized Approach

Kristóf Bérczi^a Alexander Göke^b
Lydia Mirabel Mendoza-Cadena^c Matthias Mnich^d

January 2022

Abstract

Deciding feasibility of large systems of linear equations and inequalities is one of the most fundamental algorithmic tasks. However, due to data inaccuracies or modeling errors, in practical applications one often faces linear systems that are *infeasible*. Extensive theoretical and practical methods have been proposed for post-infeasibility analysis of linear systems. This generally amounts to detecting a feasibility blocker of small size k , which is a set of equations and inequalities whose removal or perturbation from the large system of size m yields a feasible system. This motivates a parameterized approach towards post-infeasibility analysis, where we aim to find a feasibility blocker of size at most k in fixed-parameter time $f(k) \cdot m^{\mathcal{O}(1)}$.

We establish parameterized intractability (W[1] and NP-hardness) results already in very restricted settings for different choices of the parameters maximum size of a deletion set, number of positive/negative right-hand sides, treewidth, pathwidth and treedepth. Additionally, we rule out a polynomial compression for MINFB parameterized by the size of a deletion set and the number of negative right-hand sides.

Furthermore, we develop fixed-parameter algorithms parameterized by various combinations of these parameters when every row of the system corresponds to a difference constraint. Our algorithms capture the case of DIRECTED FEEDBACK ARC SET, a fundamental parameterized problem whose fixed-parameter tractability was shown by Chen et al. (STOC 2008).

Keywords: Directed feedback arc set, Fixed-parameter algorithms, Infeasible systems, Negative cycles

^aMTA-ELTE Momentum Matroid Optimization Research Group and MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös Loránd University, Budapest, Hungary. Email: kristof.berczi@ttk.elte.hu.

^bHamburg University of Technology, Hamburg, Germany. Email: alexander.goeke@tuhh.de

^cMTA-ELTE Momentum Matroid Optimization Research Group and MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös Loránd University, Budapest, Hungary. Email: lmendoza@protonmail.com.

^dHamburg University of Technology, Hamburg, Germany. Email: matthias.mnich@tuhh.de. Preliminary versions of the results of the paper appeared in [30, 29].

1 Introduction

Linear programming is without doubt one of the most powerful tools of optimization theory. However, the data that is used in these systems may be subject to inaccuracies and uncertainties, and therefore may lead to systems which are infeasible. Infeasibility may also be the result of modelling errors, or simply incompatibility of constraints. However, infeasibility itself allows for little conclusions; for a large system of millions of inequalities, infeasibility may stem from a very small subset of data. A natural question is therefore to detect the smallest number of changes which must be made to a given system in order to make it feasible. The analysis of infeasible linear systems has been extensively investigated [1, 2, 4, 14, 15, 16, 37, 41]; we refer to the book by Chinneck [16] for an overview.

Formally, the MINIMUM FEASIBILITY BLOCKER (MINFB) problem takes as input a system \mathcal{S} of linear inequalities $Ax \leq b$ and asks for a smallest subset \mathcal{I} such that $\mathcal{S} \setminus \mathcal{I}$ is *feasible*. As \mathcal{I} “blocks” the feasibility of \mathcal{S} , we refer to \mathcal{I} as a *feasibility blocker*; we avoid calling \mathcal{I} a “solution”, to avoid confusion with the solution of the linear system $\mathcal{S} \setminus \mathcal{I}$. Instead of removing the set \mathcal{I} of inequalities, we can equivalently perturb the right-hand sides $b_{\mathcal{I}}$ of the inequalities in \mathcal{I} ; that is, we can increase the b -values of inequalities in \mathcal{I} to a value such that the perturbed system becomes feasible.

When talking about feasibility, we have to specify over which field, and our choice here is the field \mathbb{Q} . Over this field, MINFB is NP-hard [44]; as the feasibility of a linear system can be tested in polynomial time (e.g., by the ellipsoid method [31]) the MINFB problem is NP-complete. Due to its importance, the MINFB problem has been thoroughly investigated from several different viewpoints, including approximation algorithms [1, 37], polyhedral combinatorics [41], heuristics [14], mixed-integer programming [18], and hardness of approximation [2].

Here we take a new perspective on the MINFB problem, based on parameterized complexity. In parameterized complexity, the problem input of size n is additionally equipped with one or more integer parameters k and one measures the problem complexity in terms of both n and k . The goal is to solve such instances by *fixed-parameter algorithms*, which run in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f . The motivation is that fixed-parameter algorithms can be practical for small parameter values k even for inputs of large size n , provided that the function f exhibits moderate growth. This is in contrast with algorithms that require time $n^{f(k)}$, which cannot be presumed to be practical for large input sizes n . To show that such impractical run times are best possible, a common approach is to show the problem to be W[1]-hard; a standard hypothesis in parameterized complexity is that no W[1]-hard problem admits a fixed-parameter algorithm. For background on parameterized complexity, we refer to the monograph by Cygan et al. [21].

For the MINFB problem, arguably the most natural parameter is the minimum size k of a feasibility blocker \mathcal{I} . The motivation for this choice is that in applications, we are interested in *small* feasibility blockers \mathcal{I} ; e.g., Chakravarti [11] argues that a feasibility blocker “with too large cardinality may be hard to comprehend and may not be very useful for post-infeasibility analysis.” Guillemot [32] explicitly posed the question of resolving the parameterized complexity of MINFB; he conjectured that

the problem is fixed-parameter tractable parameterized by the size of a minimum feasibility blocker for matrices with at most 2 non-zero entries per row.

A motivation for our approach comes from the fact that MINFB captures one of the most important problems in parameterized complexity, namely DIRECTED FEEDBACK ARC SET (DFAS): given a directed graph G , decide if G admits an arc set F of size at most k such that $G - F$ is an acyclic directed graph (DAG). It was a long-standing open question whether DFAS admits a fixed-parameter algorithm parameterized by the size k of the smallest directed feedback arc set, until Chen et al. [13] gave an algorithm with run time $4^k k! n^{\mathcal{O}(1)}$. The currently known fastest algorithm for DFAS runs in time $4^k k! k \cdot \mathcal{O}(n + m)$, and is due to Lokshtanov et al. [38]. It is not difficult to give a parameter-preserving reduction from DFAS to MINFB: for every arc (u, v) of the digraph G that serves as an input to DFAS, we add the inequality $x_u - x_v \leq -1$ to the linear system. Directed feedback arc sets F of G are then mapped to feasibility blockers of the same size by removing the constraints corresponding to arcs in F , and vice-versa; see Section 2 for details. Note that the constraint matrix A arising this way is totally unimodular, and each row has exactly two non-zero entries, one $+1$ and one -1 . Such constraints are called *difference constraints*; testing feasibility of systems of difference constraints has been investigated extensively [42, 45] due to their practical relevance most notably in temporal reasoning. It is therefore interesting to know whether the more general MINFB problem also admits a fixed-parameter algorithm for parameter k , even for special cases like totally unimodular matrices A (where testing feasibility is easy).

Another case of interest for MINFB is when the constraint matrix A has bounded treewidth, where the treewidth of A is defined as the treewidth of the bipartite graph that originates from assigning one vertex to every row and every column of A and connecting any two vertices by an edge whose corresponding entry in A is non-zero. Fomin et al. [25] gave a fast algorithm for MINFB with constraint matrices of bounded treewidth for the setting $k = 0$. At the same time, Bonamy et al. [10] showed that DFAS—a special case of MINFB—is fixed-parameter tractable parameterized by the treewidth of the underlying undirected graph of the input digraph¹. So the questions arise whether Fomin et al.’s algorithm can be extended to arbitrary values of k , or whether Bonamy et al.’s algorithm can be extended from DFAS to MINFB.

Besides treewidth, another parameter in the context of the structure of the constraint matrix A is the treedepth. This parameter measures how many recursive vertex deletions are necessary to delete the whole graph. Chan et al. [12] used the treedepth to solve (integer) linear program formulations. Also, with the help of this parameter, Iwata et al. [33] obtained an algorithm to find a negative cycle. Recently, Ganian et al. [27] made use of the treedepth to obtain a fixed-parameter algorithm that evaluates dependency-quantified-Boolean formulas.

One of the main unresolved questions around DFAS is whether it admits a *polynomial compression*. That is, one seeks an algorithm that, given any directed graph G

¹The algorithm of Bonamy et al. [10] is stated for the vertex deletion problem, but it can be modified to work for DFAS as well. Note that the standard reduction from DFAS to the vertex deletion problem which preserves the solution size does not necessarily result in a digraph whose underlying undirected graph has bounded treewidth even if the DFAS instance has this property.

and integer k , in polynomial time computes an instance I of a decision problem Π whose size is bounded by some polynomial $p(k)$, such that G admits a feedback arc set of size at most k if and only if I is a “yes”-instance of Π . The question for a polynomial compression has been stated numerous times as an open problem [47, 22, 20, 39]; from the algorithms by Chen et al. [13] and Lokshtanov et al. [38] only an *exponential* bound on the size of I follows. On the other hand, parameterized complexity provides tools such as cross-composition to rule out the existence of such polynomials $p(k)$ modulo the non-collapse of the polynomial hierarchy; we refer to Bodlaender et al. [8] for background. Given the elusiveness of this problem, we approach the (non-)existence of polynomial compression for DFAS from the angle of the more general MINFB problem.

1.1 Related Work

In their fundamental work, Arora, Babai, Stern and Sweedyk [2] considered the problem of removing a smallest set of *equations* to make a given system of linear equations feasible over \mathbb{Q} . They gave strong inapproximability results, showing that finding *any* constant-factor approximation is NP-hard. Berman and Karpinski [4] gave the first (randomized) polynomial-time algorithm with sublinear approximation ratio for this problem.

Giannopolous, Knauer and Rote [28] considered the “dual” of MINFB from a parameterized point of view: namely, in MAXFS we ask for a largest subsystem of an n -dimensional linear system \mathcal{S} which is feasible over \mathbb{Q} . They showed that deciding whether a feasible subsystem of at least ℓ inequalities in \mathcal{S} exists is W[1]-hard parameterized by $n + \ell$, even when \mathcal{S} consists of equations only.

For systems of equations over *finite* fields, finding minimum feasibility blockers has been considered from a parameterized perspective. In particular, over the binary field \mathbb{F}_2 , Crowston et al. [19] proved W[1]-hardness even if each equation has exactly three variables and every variable appears in exactly three equations; they further gave a fixed-parameter algorithm for the case where each equation has at most two variables.

1.2 Our Results

The main contribution of the paper is settling the complexity of MINFB in systems $Ax \leq b$ consisting of difference constraints for various choices and combinations of parameters, see Table 1 for an overview. Throughout the paper, we denote the number of positive and negative coordinates of b by w_+ and w_- , respectively. For non-trivial instances, $k < w_-$ holds as otherwise we can delete all constraints with negative right hand side to get a feasible system. A constraint matrix A with difference constraints can be naturally identified with a directed graph G , whose treewidth, pathwidth and treedepth is denoted by $\text{tw}(G)$, $\text{pw}(G)$ and $\text{td}(G)$, respectively.

The results in Table 1 are split horizontally by the type of the right hand side ($b \in \mathbb{Z}$ vs. $b \in \{-1, 0, 1\}$). In both cases, the rows are indexed by parameters $\{-, k, w_-\}$, where $-$ means that no parameter of this set is taken. Meanwhile, the columns are doubly indexed by parameters in $\{-, \text{tw}(G), \text{pw}(G), \text{td}(G)\}$ and $\{-, w_+\}$.

Table 1: Complexity landscape of MINFB for difference constraints. Columns of $\text{pw}(G)$ and $\text{tw}(G)$ are merged as they share the same algorithmic and hardness results. That is, algorithmic results hold already when the parameter $\text{tw}(G)$ is used, and hardness results hold already when the parameter $\text{pw}(G)$ is used. Complexities of cells with light shade are implied by cells with darker shade.

		–		pw(G) or tw(G)		td(G)	
		–	w_+	pw / tw	pw / tw + w_+	td	td + w_+
$b \in \mathbb{Z}$	–	NP-hard	NP-hard	NP-hard	NP-hard Thm. 4.3	W[1]-hard	W[1]-hard
	k	W[1]-hard	W[1]-hard	W[1]-hard	W[1]-hard Thm. 4.10	FPT Thm. 3.4	FPT
	w_-	W[1]-hard	FPT Thm. 3.9	W[1]-hard Thm. 4.15	FPT	FPT	FPT
$b \in \{\pm 1, 0\}$	–	NP-hard [44]	NP-hard Thm. 4.1	W[1]-hard	FPT Thm. 3.18	FPT Thm. 3.19	FPT
	k	W[1]-hard	Open	W[1]-hard Thm. 4.12	FPT	FPT	FPT
	w_-	Open	FPT	FPT Thm. 3.17	FPT	FPT	FPT

As $\text{tw}(G) \leq \text{pw}(G) \leq \text{td}(G)$ holds and $k \leq w_-$ for non-trivial instances, it is sufficient to obtain algorithmic or hardness results for certain combinations of the parameters. For example, a hardness result for $\text{pw}(G) + w_+ + k$ immediately translates into a hardness result for w_- .

The paper is organized as follows. In Section 2, we introduce basic definitions and discuss the connection between the MINFB and DFAS problems. Algorithms for the different combinations of parameters are presented in Section 3. While two of the problems remain open when the right-hand side of the system has the form $b \in \{-1, 0, 1\}$, we obtain fixed-parameter algorithms for these cases under the stronger assumption $b \in \{-1, 1\}$. Hardness results are discussed in Section 4, where we also rule out a polynomial compression for MINFB parameterized by $k + w_-$.

2 Preliminaries

Directed graphs and feasible potentials. Throughout, we work with finite and loopless directed graphs G with vertex set $V(G)$ and arc set $A(G)$. The sets of edges *leaving* and *entering* a vertex $v \in V(G)$ are denoted by $\delta^+(v)$ and $\delta^-(v)$, respectively. A *walk* W in G is a sequence of vertices $W = (v_0, v_1, \dots, v_\ell)$ such that $(v_i, v_{i+1}) \in A(G)$ for $i = 0, \dots, \ell - 1$. We call ℓ the *length* of the walk. A walk is *closed* if $v_0 = v_\ell$. We call the walk a *path* if all of its vertices are distinct. If all of its vertices are distinct except v_0 and v_ℓ , we call it a *cycle*. For two walks W, R , where the last vertex of W equals the first vertex of R , we denote by $W \circ R$ the *concatenation* of W and R , which is the sequence of all vertices in W followed by all vertices in R except the first.

Our directed graphs G are often *arc-weighted*, which means that they come with arc weights $w : A(G) \rightarrow \mathbb{Q}$. The sets of arcs with negative, zero, and positive weights are denoted A_- , A_0 and A_+ , respectively. Then the set of non-zero arcs is $A_{\neq 0} = A_- \cup A_+$. The numbers of negative and positive arcs are denoted by w_- and w_+ , respectively. The *weight* of a cycle C in G is then equal to the sum of its arc weights. In that spirit, we call a cycle *negative* (*non-negative*, *positive*) if its weight is *negative* (*non-negative*, *positive*). A *shortest* path or cycle is a path or cycle of minimum length. Note that “shortest” does not refer to the weight of a cycle.

We define the arc-weighted variant of the DFAS problem.

NEGATIVE DIRECTED FEEDBACK ARC SET (NEGATIVE DFAS)

Input: A directed graph G , a weight function $w : A(G) \rightarrow \mathbb{Q}$ and an integer k .

Task: Find a set $X \subseteq A(G)$ of size at most k such that $G - X$ has no negative cycles.

A solution X to the NEGATIVE DFAS problem is called a *negative directed feedback arc set* of (G, w) . We will often use the notion of feasible potentials.

Definition 2.1. Let (G, w) be an arc-weighted directed graph. A vertex function $\pi : V(G) \rightarrow \mathbb{R}$ is called *feasible potential* if for all arcs $a = (u, v) \in A(G)$ we have $\pi(u) - \pi(v) + w(a) \geq 0$.

The fundamental result of Gallai [26] provides a connection between feasible potentials and negative cycles.

Proposition 2.2 (Gallai [26]). *An arc-weighted directed graph (G, w) contains no cycle of negative total weight if and only if it has a feasible potential.*

By Proposition 2.2, for every solution S of a NEGATIVE DFAS instance (G, w, k) there is a feasible potential π_S on $G - S$. We often call a feasible potential of $G - S$ a *feasible potential for S* . Such a feasible potential also certifies that S is indeed a solution.

Let us now formally define the MINFB problem.

MINIMUM FEASIBILITY BLOCKER (MINFB)

Input: A coefficient matrix $A \in \mathbb{R}^{m \times n}$, a right-hand side vector $b \in \mathbb{R}^m$ and an integer k .

Task: Find a set $\mathcal{I} \subseteq \{1, \dots, m\}$ of size at most k such that $(a_{i,\bullet} \cdot x \leq b_i)_{i \in \{1, \dots, m\} \setminus \mathcal{I}}$ is feasible.

By multiplying rows of A and the corresponding entries of b with (-1) , the MINFB problem also covers, in a parameter-equivalent way, those cases where some inequali-

ties are of the type $a_{i,\bullet} \cdot x \geq b_i$. Furthermore, equations of the form $a_{i,\bullet} \cdot x = b_i$ can be written as two inequalities $a_{i,\bullet} \cdot x \leq b_i$ and $-a_{i,\bullet} \cdot x \leq -b_i$ (equivalent to $a_{i,\bullet} \cdot x \geq b_i$). In a feasible solution x^* violating such an equation, at most one of the above inequalities is violated. Thus, the MINFB problem with equations can be reduced to the formulation of the MINFB as formulated above without changing the parameter.

Throughout the paper, we concentrate on instances of MINFB consisting of difference constraints without explicitly mentioning it, that is, each row of the matrix A contains exactly two non-zero entries, one $+1$ and one -1 . The next theorem shows that in such cases MINFB can be reformulated as a NEGATIVE DFAS instance.

Theorem 2.3. *The NEGATIVE DFAS problem and the MINFB problem for difference constraints are parameter-equivalent.*

Proof. Let (G, w, k) be a NEGATIVE DFAS instance, and let $n = |V(G)|$ and $m = |A(G)|$. Fix an arbitrary order v_1, \dots, v_n of the vertices and e_1, \dots, e_m of the arcs. Let $A = (a_{i,j}) \in \mathbb{R}^{m \times n}$ denote the incidence matrix of G with entries $a_{i,j} = +1$ for $e_i \in \delta^-(v_j)$, $a_{i,j} = -1$ for $e_i \in \delta^+(v_j)$, and $a_{i,j} = 0$ otherwise. Furthermore, let $b_i = w(\alpha_i)$. The resulting tuple (A, b, k) is an instance of the MINFB problem with A being a matrix of difference constraints.

The construction is bijective in the following sense. Given an instance (A, b, k) of MINFB with $A = (a_{i,j}) \in \mathbb{R}^{m \times n}$ consisting of difference constraints, define a directed graph on n vertices v_1, \dots, v_n as follows. For every constraint $a_{i,\bullet} \cdot x \leq b_i$, we add an arc $e = (v_{j^+}, v_{j^-})$, where j^- is the unique index with $a_{i,j^-} = -1$ and j^+ is the unique index with $a_{i,j^+} = +1$. Furthermore, set the weight of e to be $w(e) = b_i$. Let G denote the resulting digraph. Then (G, w, k) is a NEGATIVE DFAS instance.

Our goal is to compare the solutions of the two problems. Intuitively, deleted constraints and arcs are in a one to one correspondence. Formally, for each $X \subseteq A(G)$ denote by I_X the corresponding indices of the constraints. Then we get the following chain of equivalent statements:

- $(G - X, w)$ contains no negative cycles with respect to w .
- $\Leftrightarrow (G - X, w)$ has a feasible potential $\pi : V(G) \rightarrow \mathbb{R}$.
- \Leftrightarrow There is some $\pi : V(G) \rightarrow \mathbb{R}$ such that $\pi(u) \leq \pi(v) + w(e)$ for all $e = (u, v) \in A(G) \setminus X$.
- \Leftrightarrow There is some $x \in \mathbb{R}^{V(G)}$ such that $x_u - x_v \leq w(e)$ for all $e = (u, v) \in A(G) \setminus X$.
- \Leftrightarrow There is some $x \in \mathbb{R}^n$ such that $a_{i,\bullet} \cdot x \leq b_i$ for all $i \in \{1, \dots, m\} \setminus I_X$.

Furthermore, as X and $X_{\mathcal{I}}$ have the same cardinality, we get that X is a solution to (G, w, k) if and only if $X_{\mathcal{I}}$ is a solution to (A, b, k) . \square

Observe that all instances of NEGATIVE DFAS with number of negative arcs smaller or equal to k are solvable in polynomial time.

Theorem 2.4. *Let (G, w, k) be a NEGATIVE DFAS instance with $w_- \leq k$. Then (G, w, k) has a solution. Moreover, we can check for this condition and compute a solution in linear time.*

Proof. Iterate over $A(G)$ and collect the set A_- of all arcs with negative weight. Then $G - A_-$ contains no negative cycles, as it contains no arcs of negative weight. Moreover, we can check whether $|A_-| = w_- \leq k$ and possibly return A_- as a solution. \square

Treewidth, pathwidth and treedepth. For an undirected graph H , a *tree decomposition* (*path decomposition*) is a pair (T, \mathcal{B}) where T is a tree (path) and \mathcal{B} a collection of bags $B_v \subseteq V(H)$, each bag corresponding to some node $v \in V(T)$. The bags have the property that for any edge of H , there is a bag in \mathcal{B} that contains both of its endpoints. Furthermore, the bags containing v form a subtree (subpath) of T for each vertex $v \in V(H)$. The *width* of (T, \mathcal{B}) is defined as the largest bag size of \mathcal{B} minus one. The *treewidth* (*pathwidth*) of H is the minimum width $\text{tw}(G)$ over all tree (path) decompositions of H . For a directed graph G the treewidth (pathwidth) of G is defined as the treewidth (pathwidth) of the underlying undirected graph of G .

Nice tree decompositions are useful structures for dynamic programs in graphs of bounded treewidth.

Definition 2.5. A *nice tree decomposition* is a rooted tree decomposition $(T, \{B_i\}_{i \in V(T)})$ of a graph H with the following properties:

- For the root r and for every leaf ℓ , $B_r = \emptyset = B_\ell$.
- Each non-leaf node x of T belongs to one of the following categories:
 - *Introduce node.* The node x has exactly one child y so that $B_x = B_y \cup \{v\}$ for some vertex $v \in V(H)$.
 - *Forget node.* The node x has exactly one child y so that $B_x = B_y \setminus \{v\}$ for some vertex $v \in V(H)$.
 - *Join node.* The node x has exactly two children y and y' so that $B_x = B_y = B_{y'}$.

For $x \in V(G)$, we denote by T_x the subtree of T rooted at x . Moreover, we define G_x to be the graph $G[\bigcup_{y \in T_x} B_y]$.

Given a tree decomposition with width $\text{tw}(G)$, we can obtain a nice tree-decomposition with width $\text{tw}(G)$ in time $\mathcal{O}(n)$, see [35, Lemma 13.1.3]. Thus, by a result of Bodlaender [5], such a nice tree decomposition can be determined efficiently.

Proposition 2.6 (Bodlaender [5]). *Given a graph G with n vertices, there is an algorithm that constructs a nice tree decomposition of size $\text{tw}(G)$ with $\mathcal{O}(n)$ nodes in $2^{\text{tw}(G)^3} n$ time.*

For our usage, it is enough to have a nice tree decomposition that is within a constant factor of minimum width. This allows us to use the following faster algorithm by Bodlander et al. [6].

Proposition 2.7 ([6]). *Given a graph G with n vertices, there is an algorithm that constructs a nice tree decomposition of size $5\text{tw}(G) + 4$ with $\mathcal{O}(n)$ nodes in $2^{\mathcal{O}(\text{tw}(G))}n$.*

The *treedepth* of an undirected graph can be defined in different ways, usually taking a certain decomposition of the graph. We follow the definition given by Nešetřil and Ossona de Mendez [40].

Definition 2.8. Let G be an undirected graph with connected components G_1, \dots, G_q . Then the *treedepth* $\text{td}(G)$ of G is

$$\text{td}(G) = \begin{cases} 1, & \text{if } |V(G)| = 1; \\ 1 + \min_{v \in V(G)} \text{td}(G - v), & \text{if } q = 1 \text{ and } |V(G)| > 1; \\ \max\{\text{td}(G_i) : G_i \text{ is a connected component of } G\}, & \text{otherwise.} \end{cases}$$

The *treedepth* of a directed graph is the treedepth of its underlying undirected graph.

Bodlaender et al. have shown relations between treewidth, pathwidth and treedepth.

Proposition 2.9 (Bodlaender et al. [7]). *It holds $\text{tw}(G) \leq \text{pw}(G) \leq \text{td}(G) - 1$ for any graph G .*

Thus, the class of bounded treewidth graphs contains the class of bounded pathwidth graphs, which in turn contains the class of bounded treedepth graphs.

Parameterized complexity. For our hardness results we will use two different kinds of hardness. The first one is $\text{W}[1]$ -hardness which, under the standard assumption $\text{W}[1] \neq \text{FPT}$, implies that there is no fixed-parameter algorithm for problems of this type. The other hardness considers compressions. A *polynomial compression* of a language L into a language Q is a polynomial-time computable mapping $\Phi : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^*$, $\Phi((x, k)) \mapsto y$ such that $((x, k) \in L \Leftrightarrow y \in Q)$ and $|y| \leq k^{\mathcal{O}(1)}$ for all $(x, k) \in \Sigma^* \times \mathbb{N}$. Many natural parameterized problems do not admit polynomial compressions under the hypothesis $\text{NP} \not\subseteq \text{coNP}/\text{poly}$.

Both types of hardness can be transferred to other problems by “polynomial parameter transformations”, which were first proposed by Bodlaender et al. [9].

Definition 2.10. Let Σ be an alphabet. A *polynomial parameter transformation (PPT)* from a parameterized problem $\Pi \subseteq \Sigma^* \times \mathbb{N}$ to a parameterized problem $\Pi' \subseteq \Sigma^* \times \mathbb{N}$ is a polynomial-time computable mapping $\Phi : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N}$, $(x, k) \mapsto (x', k')$, such that $k' = k^{\mathcal{O}(1)}$, and $(x, k) \in \Pi \Leftrightarrow (x', k') \in \Pi'$ for all $(x, k) \in \Sigma^* \times \mathbb{N}$. Two parameterized problems are *parameter-equivalent* if there are PPTs in both directions and the transformations additionally fulfill that $k' = k$.

Note that polynomial parameter transformations are transitive. Further, a PPT from Π to Π' together with a polynomial compression for Π' yields a polynomial compression for Π . This can be used to rule out polynomial compressions.

Proposition 2.11 (Kratsch and Wahlström [36]). *Let Π, Π' be parameterized problems. If there is a polynomial parameter transformation from Π to Π' and Π admits no polynomial compression, then neither does Π' .*

As mentioned in the introduction, the complexity class containing all fixed-parameter tractable problems is called FPT. Thus, on the one hand, if a parameterized problem (P, k) is in FPT, then the parameterized problem $(P, k + k')$ is also in FPT for any parameter k' . Conversely, on the other hand, if a parameterized problem $(P, k + k')$ is W[1]-hard, then the parameterized problem (P, k) is also W[1]-hard.

3 Fixed-Parameter Algorithms for Finding Minimum Feasibility Blockers

In this section, we develop fixed-parameter algorithms for different combinations of the parameters maximum size k of a deletion set (k), number of positive/negative right-hand sides (w_+, w_-), treewidth/pathwidth ($\text{tw}(G), \text{pw}(G)$) and treedepth ($\text{td}(G)$). Recall that we always assume that the constraint matrix is of difference constraints. By Theorem 2.3, it suffices to consider the corresponding NEGATIVE DFAS problem instead, therefore we use the directed graph terminology.

Before stating the algorithms, we give an overview how the feasibility of a set $S \subseteq A(G)$ can be checked, that is, how to decide if $G - S$ has indeed no negative cycles. For this, we use the Floyd–Warshall algorithm [23, 46, 43] that finds shortest paths between all pairs of vertices in a weighted digraph with no negative cycles. The algorithm can additionally detect whether there is a negative cycle in the directed graph and outputs a shortest one if exists.

Proposition 3.1 (Floyd [23], Roy [43], Warshall [46]). *Given an arc-weighted directed graph (G, w) we can detect in time $\mathcal{O}(n^3)$ whether G contains a negative cycle. Moreover, we can recover in the same time a shortest (i.e. length-minimal) negative cycle if one exists.*

Running this algorithm on $(G - S, w|_{A(G) \setminus S})$ and additionally checking whether $|S| \leq k$, allows us to verify a solution to some NEGATIVE DFAS instance (G, w, k) .

3.1 Algorithm for Bounded Treedepth and Solution Size

The following result, showed by Nešetřil and Ossona de Mendez [40], exhibits that all cycles of a graph with bounded treedepth have bounded length.

Lemma 3.2. *Let G be a graph, P be a path in G , and C be a cycle of G . Then $|P| \leq 2^{\text{td}(G)} - 1$ and $|C| \leq 2^{\text{td}(G)-1}$ holds.*

The high-level idea is then to detect negative cycles iteratively in the directed graph, and to branch on the arcs of the cycle. We now present the main algorithm of this section. The run-time depends on the maximum length of a negative cycle in the given graph. We will revisit it later on, when we derive other bounds on the length of negative cycles in G depending on the parameter choices.

Lemma 3.3. *There is an algorithm solving any NEGATIVE DIRECTED FEEDBACK ARC SET instance (G, w, k) in time $\mathcal{O}(L^k n^3)$, where L is an upper bound on the length of any negative cycle in G .*

Proof. We recursively call the following procedure with some potential partial solution $S \subseteq V(G)$ with $|S| \leq k$. The initial call is done with $S = \emptyset$. First, we check in time $\mathcal{O}(n^3)$ whether there is a negative cycle C in $G - S$ and recover it via Proposition 3.1. If there is no negative cycle and $|S| \leq k$, we return S as a solution. If there is a negative cycle C and $|S| = k$, we give up on this branch. Otherwise, there is a negative cycle C and $|S| < k$. Then for every $v \in V(C)$ we make a subroutine call with $S_v = S \cup \{v\}$. If our initial call with $S = \emptyset$ did not return a solution for any branch, we report that the instance has no solution. This finishes the description of our algorithm.

As we only make subroutine calls in the case of $|S| < k$, we add only one vertex to S , and we start with $|\emptyset| = 0 \leq k$, one can show by induction that all our calls indeed fulfill $|S| \leq k$. Now we argue for the correctness. To this end, we have to show that if there is a solution, we do indeed return a solution. As we only return sets S such that $|S| \leq k$ and $G - S$ has no negative cycles, we just have to make sure that we return a set. We consider the variant of our algorithm where we do not return a solution early, but rather save it and return it at the end of the algorithm. Let S^* be an inclusionwise minimal solution to (G, w, k) . We are going to reconstruct a possible subroutine call sequence $\emptyset = S_0 \subsetneq S_1 \subsetneq \dots \subsetneq S_{|S^*|} = S^*$ that is called by our algorithm.

We prove by induction on i that S_i appears in one branch of our algorithm. We start with $S_0 = \emptyset$ which is our initial subroutine call. As long as $i < |S^*|$, we have that $S_i \subsetneq S^*$ and thus, by inclusion-wise minimality of S^* , $G - S_i$ contains at least one negative cycle C_i . Now S^* is a solution, so we know that there is a $v_i \in S^* \cap V(C_i)$. By C_i being disjoint from S_i , we have that $S_{i+1} = S_i \cup \{v_i\}$ is a strict superset of S_i . Thus we make a subroutine call with this S_{i+1} . This shows by induction that our modified algorithm considers the set S^* and our original algorithm returns a solution.

For the run-time, note that at each sub-routine call we branch into $|C| \leq L$ many branches. We start with $|S| = 0$ and in each recursive subroutine call $|S|$ increases by one. As we stop once $|S| = k$, this means our recursion nests at most k levels deep. This results in L^k subroutine calls. In each call, we need $\mathcal{O}(n^3)$ time to check for negative cycles. \square

The above observations enables us to give a fixed-parameter algorithm for combined parameter the treedepth and the solution size.

Theorem 3.4. *There is an algorithm solving MINFB in time $\mathcal{O}(2^{k \text{td}(G)} n^2 m)$.*

Proof. By Theorem 2.3, we can construct an instance of NEGATIVE DFAS in polynomial time that has the same parameters $\text{td} + k$ such that the original instances is a “yes”-instance if and only if the constructed instance is. The theorem then follows by combining Lemma 3.2 and Lemma 3.3. \square

3.2 Algorithm for Bounded Number of Positive and Negative Arcs

The goal of this section is to derive an algorithm for NEGATIVE DFAS parameterized in w_+ and w_- . In order to do so, we build on skew cuts in directed graphs.

Definition 3.5. Let G be a directed graph, $p \in \mathbb{Z}_{\geq 0}$ be a non-negative integer, and $X_1, \dots, X_p, Y_1, \dots, Y_p \subseteq V(G)$ be pairwise disjoint vertex sets of G . An $(X_1, \dots, X_p) \rightarrow (Y_1, \dots, Y_p)$ -skew cut is an arc set $S \subseteq A(G)$ such that there is no $X_i \rightarrow Y_j$ -path in $G - S$ for any $1 \leq j \leq i \leq p$.

Then the SKEW SEPARATOR problem is defined as follows.

SKEW SEPARATOR

Input: A graph G , vertex sets $X_1, \dots, X_p, Y_1, \dots, Y_p \subseteq V(G)$ and an integer $k \in \mathbb{Z}_{\geq 0}$.

Task: Find an $(X_1, \dots, X_p) \rightarrow (Y_1, \dots, Y_p)$ -skew cut of size at most k or decide that no such skew cut exists.

The SKEW SEPARATOR problem was introduced by Chen et al. [13], who devised an algorithm for it as a subroutine for their DIRECTED FEEDBACK VERTEX SET algorithm. In fact, they solve the vertex deletion variant, but it can be applied to the arc deletion variant by subdividing arcs.

Proposition 3.6 (Chen et al. [13]). *Any instance $(G, (X_1, \dots, X_p), (Y_1, \dots, Y_p), k)$ of SKEW SEPARATOR can be solved in time $\mathcal{O}(4^k kn^3)$.*

The high-level description of our algorithm is the following. We first guess the intersection between a solution and the set of non-zero arcs. Afterwards, we focus on the directed graph induced by zero-weight arcs only. For any solution S , the graph $G - S$ has a feasible potential, that in turn defines an ordering of the end-points of the non-zero arcs. We use this observation to show that solutions can be obtained by solving a SKEW SEPARATOR problem in the zero-arc graph. As we do not know the feasible potential or the exact SKEW SEPARATOR instance, our algorithm guesses among all potential orderings of the endpoints of the non-zero arcs.

The main graph we work with is constructed from G by removing the arcs of $A_{\neq 0}$ and splitting their endpoints.

Definition 3.7. Let (G, w) be an arc-weighted directed graph and denote by Z the set of vertices $z \in V(G)$ with $(\delta^+(z) \cup \delta^-(z)) \cap A_{\neq 0}(G) \neq \emptyset$. The *zero-weight propagation graph* of G is the graph \vec{G}_0 obtained from G by deleting $A_{\neq 0}$ and splitting every $z \in Z$ into two vertices z^+ and z^- , where z^+ inherits the outgoing arcs and z^- inherits the incoming arcs of z . In this context we denote for every subset $Y \subseteq Z$ by Y^+ and Y^- the set of all z^+ 's and z^- 's with $z \in Y$, respectively.

By abuse of notation, we will identify the arcs in \vec{G}_0 with those of A_0 . We are now ready state the main observation of this chapter.

Lemma 3.8. *Let (G, w) be an arc-weighted directed graph and let $S \subseteq A_0(G)$ be a negative directed feedback arc set of (G, w) . Let Z be the endpoints of arcs in $A_{\neq 0}(G)$ and let \vec{G}_0 be the zero-weight propagation graph of G . Then there is an ordered partition (Z_1, \dots, Z_p) of Z such that*

1. S is a $(Z_1^+, \dots, Z_p^+, \emptyset) \rightarrow (\emptyset, Z_1^-, \dots, Z_p^-)$ -skew cut in \vec{G}_0 , and
2. any $(Z_1^+, \dots, Z_p^+, \emptyset) \rightarrow (\emptyset, Z_1^-, \dots, Z_p^-)$ -skew cut in \vec{G}_0 is a negative directed feedback arc set of (G, w) .

Proof. As S is a negative directed feedback arc set of (G, w) , the graph $G' = G - S$ has a feasible potential π with respect to $w|_{A(G')}$. We define an ordered partition (Z_1, \dots, Z_p) of Z by using the potential π . Two vertices z and z' belong to the same partition class if and only if $\pi(z) = \pi(z')$. This way each partition class admits a unique value that is possessed by all of its elements as a potential. The partition classes are then ordered according to these values in a decreasing order.

We now verify the statements of the theorem. First we check, whether S is a $(Z_1^+, \dots, Z_p^+, \emptyset) \rightarrow (\emptyset, Z_1^-, \dots, Z_p^-)$ -skew cut in \vec{G}_0 . Suppose, for sake of contradiction, that this does not hold. Then there is an $x^+ \rightarrow y^-$ -path in \vec{G}_0 with $x^+ \in Z_i^+$ and $y^- \in Z_j^-$ such that $j < i$. By the ordering of the partition classes, we get $\pi(y) > \pi(x)$. Now consider the corresponding $x \rightarrow y$ -path P in $G - S$, where x and y are the original vertices of x^+ and y^- before they got split. By construction of \vec{G}_0 , P contains arcs of weight zero only. As π is a feasible potential, $\pi(u) - \pi(v) = \pi(u) - \pi(v) + w(a) \geq 0$ for each arc $a = (u, v)$ of P . Summing this up over all arcs in P , we get $\pi(x) - \pi(y) = \sum_{a=(u,v) \in A(P)} \pi(u) - \pi(v) \geq 0$. In other words $\pi(y) \leq \pi(x)$, a contradiction.

It remains to verify that any $(Z_1^+, \dots, Z_p^+, \emptyset) \rightarrow (\emptyset, Z_1^-, \dots, Z_p^-)$ -skew cut in \vec{G}_0 is a negative directed feedback arc set of (G, w) . Let S^* be such a skew cut. We try to prove that S^* is a negative directed feedback arc set of (G, w) by constructing a feasible potential π^* of $G^* = G - S^*$ with respect to $w|_{A(G^*)}$. We may assume that $\max_{v \in V(G)} \pi(v) \leq 0$ as the potentials can be shifted by the same value on every vertex. We define π^* as

$$\pi^*(v) = \begin{cases} \pi(v) & , \text{ if } v \in Z \\ \min\{\pi(z) \mid \text{there is a } z^+ \rightarrow v\text{-path in } \vec{G}_0 - S^*\} & , \text{ otherwise} \end{cases}$$

where we define the minimum to be 0 if there is no $z^+ \rightarrow v$ -path in $\vec{G}_0 - S^*$ for any $z^+ \in Z^+$.

We show that π^* is a feasible potential in G^* . As $A_{\neq 0}(G^*) = A_{\neq 0}(G)$, $u, v \in Z$ holds for each arc $a = (u, v) \in A_{\neq 0}(G^*)$. This implies $\pi^*(u) - \pi^*(v) + w(a) = \pi(u) - \pi(v) + w(a) \geq 0$. Hence, it remains to check arcs $a = (u, v) \in A_0(G^*) = A_0(G) \setminus S^*$. If $u \in Z$, then the arc (u^+, v^-) or (u^+, v) exists in $\vec{G}_0 - S^*$. In the former case we again have $u, v \in Z$ and the potential is feasible for this arc. Otherwise, we have that

there is an $u^+ \rightarrow v$ -path in $\vec{G}_0 - S^*$ and thus $\pi^*(v) \leq \pi(u) = \pi^*(u)$. As $w(a) = 0$, this implies $\pi^*(u) - \pi^*(v) + w(a) \geq 0$. So we can assume that $u \notin Z$. If additionally u does not have a $z^+ \rightarrow u$ -path in $\vec{G}_0 - S^*$, then $\pi^*(u) = 0$ and by $\max_{v \in V(G)} \pi(v) \leq 0$, we have that $\pi^*(v) \leq 0 = \pi^*(u)$, implying $\pi^*(u) - \pi^*(v) + w(a) \geq 0$. Now we know that $u \notin Z$ and there is a $z^+ \rightarrow u$ -path in $\vec{G}_0 - S^*$ for some $z \in Z$. We distinguish between the two cases $v \notin Z$ and $v \in Z$. If $v \notin Z$, then the arc (u, v) exists in $\vec{G}_0 - S^*$. Hence, any $z^+ \rightarrow u$ -path can be prolonged to a $z^+ \rightarrow v$ -path by adding (u, v) at the end. Thus $\pi^*(v) \leq \pi^*(u)$, and $\pi^*(u) - \pi^*(v) + w(a) \geq 0$. If $v \in Z$, we have that the arc (u, v^-) exists in $\vec{G}_0 - S^*$. Any $z^+ \rightarrow u$ -path can be prolonged to a $z^+ \rightarrow v^-$ -path. By definition of our SKEW SEPARATOR instance, we have that for all such paths in $\vec{G}_0 - S^*$, we have $\pi(z) \geq \pi(v)$. This implies $\pi^*(v) \leq \pi^*(u)$, and so $\pi^*(u) - \pi^*(v) + w(a) \geq 0$. Hence, π^* is a feasible potential for $G - S^*$ and thus S^* is a negative directed feedback arc set of (G, w) . \square

We are now going to use this result to give an algorithm solving the NDFAS problem parameterized by w_+ and w_- , and thus we solve our original problem.

Theorem 3.9. *There is an algorithm that solves a MINFB instance (G, w, k) in time $2^{\mathcal{O}((w_+ + w_-) \log(w_+ + w_-))} n^{\mathcal{O}(1)}$.*

Proof. By Theorem 2.3, we can construct an instance of NEGATIVE DFAS in polynomial time that has the same parameters $w_+ + w_-$ such that the original instance is a “yes”-instance if and only if the constructed instance is.

If $w_- \leq k$, return the set $A_-(G)$ of negative arcs. Otherwise, for every subset $A' \subseteq A_{\neq 0}(G)$ of size at most k do the following. For the graph $G' = G - A'$ with Z' being endpoints of non-zero arcs $A_{\neq 0}(G')$ and for every ordered partition (Z_1, \dots, Z_p) of Z' , we call the algorithm for SKEW SEPARATOR (see Proposition 3.6) on

$$(\vec{G}'_0, (Z_1^+, \dots, Z_p^+, \emptyset), (\emptyset, Z_1^-, \dots, Z_p^-), k - |A'|).$$

If the algorithm returns a set S' , we check whether $A' \cup S'$ is a negative directed feedback arc set for G and if it is, we return it. If the algorithm does not find a negative directed feedback arc set for any A' and ordered partition (Z_1, \dots, Z_p) of Z' , we return that the problem has no negative directed feedback arc set of size at most k .

As we check any solution we return for correctness, we just have to verify that we always return a negative directed feedback arc set, if there is any. Let S be a negative directed feedback arc set for (G, w) . For $A' = S \cap A_{\neq 0}(G)$ we know that $S' = S \setminus A'$ is a negative directed feedback arc set for $G' = G - A'$, as $(G - A') - S' = G - S$ which contains no negative cycles. Now $S' \cap A_{\neq 0}(G') = \emptyset$, and thus by Lemma 3.8 there is an ordered partition (Z_1, \dots, Z_p) of Z' such that S' is a $(Z_1^+, \dots, Z_p^+, \emptyset) \rightarrow (\emptyset, Z_1^-, \dots, Z_p^-)$ -skew cut in \vec{G}'_0 . Thus, we know that $(\vec{G}'_0, (Z_1^+, \dots, Z_p^+, \emptyset), (\emptyset, Z_1^-, \dots, Z_p^-), k - |A'|)$ is a “yes”-instance. Moreover, Lemma 3.8 tells us that any solution S^* to this SKEW SEPARATOR instance our algorithm call finds, is a negative directed feedback arc set for G' . Thus, $A' \cup S^*$ is a negative directed feedback arc set of G , as $G - (A' \cup S^*) = G' - S^*$.

Also, $A' \cup S^*$ has size at most k . Hence, our algorithm finds a negative directed feedback arc set of the given size if one exists.

For the run-time note that by returning the trivial solution A_- if $w_- \leq k$, we have either a linear run-time for constructing A_- or $k < w_-$. Now we have at most $2^{w_+ + w_-}$ possible subsets A' of $A_{\neq 0}$. The set Z' contains at most $2w_+ + 2w_-$ vertices and the number of its ordered partitions can be bounded by $(2w_+ + 2w_-)! \cdot 4^{w_+ + w_-}$ (orders of Z' times the choice for every element whether to start a new subset there). Thus, the overall run-time is

$$\mathcal{O}(2^{w_+ + w_-} \cdot (2w_+ + 2w_-)! \cdot 4^{w_+ + w_-} \cdot (4^k k n^3 + n^{\mathcal{O}(1)})),$$

which, using $k < w_-$, can be rewritten as $2^{\mathcal{O}((w_+ + w_-) \log(w_+ + w_-))}$ poly(n). \square

3.3 Number of Positive or Negative Arcs and Feasible Potentials

This section is dedicated to NEGATIVE DFAS with arc weights in $\{-1, 0, 1\}$ and how the number of positive or negative arcs influences the feasible potential of the solution. We use this results for the algorithms in the following section. We show that for both parameters w_- and w_+ , the solution must have an integral feasible potential in $[0, w_-]$ and $[0, w_+]$, respectively. Unfortunately, the latter only holds for graphs that are strongly connected after removing the solution. Moreover, we also prove that for the parameter treedepth $\text{td}(G)$, the solution has a feasible potential in $[0, 2^{\text{td}(G)}]$

We first state a folklore technique for constructing a feasible potential of a graph.

Lemma 3.10. *Let (G, w) be an arc-weighted directed graph that contains no negative cycles. Let G^* be the directed graph obtained from G by introducing a new vertex s^* that is connected to every original vertex $v \in V(G)$ by an arc (s^*, v) of weight 0. For every $v \in V(G^*)$, let $\pi^*(v)$ be the weight of a minimum-weight $s^* \rightarrow v$ -path P_v . Then $\pi = \pi^*|_{V(G)}$ is a feasible potential for G .*

Proof. We claim that π^* is a feasible potential for G^* . Suppose, for sake of contradiction, that this is not the case. Then there is an arc $a = (u, v)$ for which $\pi^*(u) - \pi^*(v) + w(a) < 0$. In other words, we have that $w(P_u \circ a) < w(P_v)$. As P_v is a minimum weight $s^* \rightarrow v$ -path, we have that $P_u \circ a$ is only an $s^* \rightarrow v$ -walk. Moreover, $P_u \circ a$ has to contain a negative closed walk, as otherwise it would contain an $s^* \rightarrow v$ -path of weight less than P_v . As P_u is a path, this closed walk is indeed a cycle O containing the arc a . Now s^* has only outgoing arcs, thus the cycle lies in $G^* - s^* = G$, a contradiction to G having no negative cycles. So π^* is a feasible potential for G^* . By $G = G^* - s^*$ being a subgraph of G^* , we get that the function $\pi = \pi^*|_{V(G)}$ is a feasible potential for G . \square

With this construction in place, we can make our observations about integral feasible potentials with few distinct values.

Lemma 3.11. *Let G be a directed graph, and $w : A(G) \rightarrow \{-1, 0, 1\}$ be a weight function. If G contains no negative cycles, then there is a feasible potential $\pi : V(G) \rightarrow [0, w_-]$.*

Proof. First, we use Lemma 3.10 to construct the auxiliary graph G^* with its function π^* . Now note that any $s^* \rightarrow v$ -path can contain at most w_- -many negative arcs and each of those has weight -1 . Thus, we have that $\pi^*(v) = w(P_v) \geq -w_-$ for all $v \in V(G^*)$. Moreover, the arc (s^*, v) always forms an $s^* \rightarrow v$ -path of weight 0. By all arc-weights being integral, we have that π^* is an integer function with values in $[-w_-, 0]$. Now, by Lemma 3.10, $\pi = \pi^*|_{V(G)}$ is a feasible potential for G , which has only integer values in $[-w_-, 0]$. Shifting all values by w_- shows the statement. \square

Lemma 3.12. *Let G be a strongly connected directed graph, and $w : A(G) \rightarrow \{-1, 0, 1\}$ be a weight function. If G contains no negative cycles, then there is a feasible potential $\pi : V(G) \rightarrow [0, w_+]$.*

Proof. First, we use Lemma 3.10 to construct the auxiliary graph G^* with its function π^* . Suppose, for sake of contradiction, that an $s^* \rightarrow v$ -path P_v has weight less than $-w_+$. Let P be the $u \rightarrow v$ -subpath of P_v that contains everything but the first arc $a = (s^*, u)$ of P_v . As $w(a) = 0$ we know that $w(P) < -w_+$ and moreover P exists in G . Now, G is strongly connected, and therefore there is a $v \rightarrow u$ -path Q in G . As G contains only w_+ -many positive arcs and all of them have weight 1, we get $w(Q) \leq w_+$. Hence, $P \circ Q$ is a closed walk of weight $w(P) + w(Q) < w_+ - w_+ = 0$. This negative closed walk contains a negative cycle, in contradiction to G having none of those. Thus, we have that $\pi^*(v) = w(P_v) \geq -w_+$ for all $v \in V(G^*)$. Moreover, the arc (s^*, v) always forms an $s^* \rightarrow v$ -path of weight 0. By all arcs weights being integral, we have that π^* is an integer function with values in $[-w_+, 0]$. Now, by Lemma 3.10, $\pi = \pi^*|_{V(G)}$ is a feasible potential for G , which has only integer values in $[-w_+, 0]$. Shifting all values by w_+ shows the theorem. \square

Lemma 3.13. *Let G be a directed graph, and $w : A(G) \rightarrow \{-1, 0, 1\}$ be a weight function. If G contains no negative cycles, then there is a feasible potential $\pi : V(G) \rightarrow [0, 2^{\text{td}(G)}]$.*

Proof. First, we use Lemma 3.10 to construct the auxiliary graph G^* with its function π^* . By Lemma 3.2 we have that any path in G contains at most $2^{\text{td}(G)}$ many arcs. Moreover, any $s^* \rightarrow v$ -path consists of an arc (s^*, u) of weight 0, followed by an $u \rightarrow v$ -path in G . Thus, any $s^* \rightarrow v$ -path can contain at most $2^{\text{td}(G)}$ many negative arcs and each of those has weight -1 . Thus, we have that $\pi^*(v) = w(P_v) \geq -2^{\text{td}(G)}$ for all $v \in V(G^*)$. Moreover, the arc (s^*, v) always forms an $s^* \rightarrow v$ -path of weight 0. By all arcs weights being integral, we have that π^* is an integer function with values in $[-2^{\text{td}(G)}, 0]$. Now, by Lemma 3.10, $\pi = \pi^*|_{V(G)}$ is a feasible potential for G , which has only integer values in $[-2^{\text{td}(G)}, 0]$. Shifting all values by $2^{\text{td}(G)}$ shows the theorem. \square

3.4 Dynamic Program for Treewidth and Bounded Feasible Potentials

The aim of this section is to apply our findings on feasible potentials with few different values to a dynamic program utilizing the treewidth. The overall approach is computing a nice treewidth decomposition of our graph (see Definition 2.5 and

Proposition 2.6) and then guessing via a dynamic program the feasible potential of some solution on each bag of the tree decomposition. The deleted arcs are then exactly those that violate the guessed potential. However, when we parameterize in the number of positive arcs, we are only guaranteed a feasible potential with few different values for every strong component of $G - S$. Therefore we also have to guess a topological order of the strong components of $G - S$ when restricted to the bags of the tree decomposition. To handle both cases simultaneously, we introduce a set \mathcal{C} which contains ordered partitions of $V(G)$ that represent components for which the guessed potential could be feasible. For the parameter w_- the single partition consisting of all vertices suffices as set \mathcal{C} . In the w_+ case, \mathcal{C} has to contain a topological order of the strong components of $G - S$. In this case we choose \mathcal{C} as all ordered partitions of $V(G)$. We generalize the properties we need for \mathcal{C} and our feasible potential to unify both choices.

Definition 3.14. Let G be a directed graph and $w : A(G) \rightarrow \mathbb{Z}$ be a weight function. We call an arc set $S \subseteq A(G)$ $((C_1, \dots, C_t), \pi)$ -feasible for some ordered partition (C_1, \dots, C_t) of $V(G)$ and some $\pi : V(G) \rightarrow \mathbb{Z}$, if for all arcs $a = (p, q) \in A(G) \setminus S$ with $p \in C_i$ and $q \in C_j$ we have either $i < j$ or $i = j$ and $\pi(p) - \pi(q) + w(a) \geq 0$.

For an ordered partition (C_1, \dots, C_t) of $V(G)$ and some $U \subseteq V(G)$, we call an ordered partition (C'_1, \dots, C'_t) of U the *projection of (C_1, \dots, C_t) on U* , denoted by $(C_1, \dots, C_t)|_U$, if for all $u \in C_i \cap C'_p$ and $v \in C_j \cap C'_q$ we have $i < j$ if and only if $p < q$. For a set \mathcal{C} of ordered partitions of $V(G)$, we denote by $\mathcal{C}|_U$ the set $\{C|_U \mid C \in \mathcal{C}\}$.

Similarly, for some $U \subseteq V(G)$ and a pair (C, π) consisting of an ordered partition C of $V(G)$ and $\pi : V(G) \rightarrow \mathbb{Z}$, we call (C', π') the *projection of (C, π) on U* for C' being the projection of C on U and $\pi' = \pi|_U$.

Let T be a tree decomposition of G with vertex bags $(B_x)_{x \in V(T)}$, i.e. a tree decomposition of the underlying undirected graph of G . We say that a set \mathcal{C} of ordered partitions of $V(G)$ is *T -compatible*, if for every $C, C' \in \mathcal{C}$ with $C|_{B_x} = C'|_{B_x}$, we have that there is a $C^* \in \mathcal{C}$, with $C^*|_{V(G_x)} = C|_{V(G_x)}$ and $C^*|_{V(G) \setminus (V(G_x) \cup B_x)} = C'|_{V(G) \setminus (V(G_x) \cup B_x)}$.

We are now able to state our general algorithm.

Lemma 3.15. *Let (G, w, k) be a NEGATIVE DFAS instance with weights $w : A(G) \rightarrow \mathbb{Z}$. Given a nice tree decomposition (T, \mathcal{B}) of G with vertex bags $(B_x)_{x \in V(T)}$ of width $\mathcal{O}(\text{tw}(G))$, a set \mathcal{C} of ordered partition of $V(G)$ compatible with T , and two integers $a \leq b$, there is an algorithm that in time $f(\mathcal{C}, T)2^{\mathcal{O}(\text{tw}(G) \log(b-a))} \cdot (n + m)$ computes a minimum size negative directed feedback arc set that is (C, π) -feasible for some $C \in \mathcal{C}$ and $\pi : V(G) \rightarrow \mathbb{Z} \cap [a, b]$. Here, $f(\mathcal{C}, T)$ is the maximum time needed to enumerate $\mathcal{C}|_{B_x}$ for any $x \in V(T)$.*

Proof. We compute a dynamic programming table D via dynamic program on $V(T)$ from the leaves upwards. The table has an entry $D[x, \kappa]$ for every $x \in V(T)$ and κ , where κ consists of an ordered partition $C \in \mathcal{C}|_{B_x}$ and an integer-valued function $\pi : B_x \rightarrow \mathbb{Z} \cap [a, b]$. The entry $D[x, \kappa]$ contains an arc set $S_{x, \kappa} \subseteq A(G_x)$ with G_x being the graph induced by the subtree decomposition of T rooted at x . Our dynamic program is motivated by the following observation:

Claim 3.16. Let $S \subseteq A(G_x)$ be a minimum-size set that is (C, π) -feasible for some $C \in \mathcal{C}|_{V(G_x)}$ and $\pi : V(G_x) \rightarrow \mathbb{Z} \cap [a, b]$. Let $((C_1^x, \dots, C_{t_x}^x), \pi^x)$ be the projection of (C, π) to B_x . Then S contains exactly those arcs $a = (p, q) \in A(G[B_x])$ with $p \in C_i^x$ and $q \in C_j^x$ such that either $j < i$ or $i = j$ and $\pi^x(p) - \pi^x(q) + w(a) < 0$.

Proof of Claim 3.16. The set S necessarily contains these arcs as otherwise it would not be (C, π) -feasible. Suppose, for sake of contradiction, that S contains further arcs. Removing such an arc from S does not violate (C, π) -feasibility and thus results in a smaller S , contradicting the minimality of S . This completes the proof of Claim 3.16. \blacksquare

The entries are computed for each $x \in V(T)$ depending on its node type.

Leaf nodes. As $B_x = \emptyset$, the only choice for κ are the empty partition $()$ and π being the empty function ε and we set $D[x, ((), \varepsilon)] = \emptyset$.

Introduce nodes. Let v be the newly introduced vertex to the bag B_x and let y be the child of x in T . Let κ consist of an ordered partition $(C_1, \dots, C_t) \in \mathcal{C}|_{B_x}$ and an integer-valued function $\pi : B_x \rightarrow \mathbb{Z} \cap [a, b]$. Define S_v to be the arcs $a = (p, q) \in \delta(v)$ with $p \in C_i$ and $q \in C_j$ such that either $j < i$ or $i = j$ and $\pi(p) - \pi(q) + w(a) < 0$. Then we set

$$D[x, \kappa] = S_v \cup D[y, \kappa'],$$

where κ' is the projection of κ to B_y .

Forget nodes. Let y be the child of x in T . Let κ consist of an ordered partition $C \in \mathcal{C}|_{B_x}$ and an integer-valued function $\pi : B_x \rightarrow \mathbb{Z} \cap [a, b]$. Denote by $K(y, \kappa)$ the set of all κ^* consisting of ordered partitions $C^* \in \mathcal{C}|_{B_y}$ and $\pi^* : B_y \rightarrow \mathbb{Z} \cap [a, b]$, whose projection to B_x is κ . Then we set

$$D[x, \pi] = \operatorname{argmin}_{\substack{S=D[y, \kappa^*] \\ \kappa^* \in K(y, \kappa)}} |S| .$$

Merge nodes. Let y_1 and y_2 the two children of x . Then for every possible choice of κ , we set

$$D[x, \kappa] = D[y_1, \kappa] \cup D[y_2, \kappa] .$$

We claim that at the root r of T , the unique entry $D[r, ((), \varepsilon)]$, where ε denotes the empty function, contains a negative directed feedback arc set for G of minimum size. First note that this is indeed the unique entry as $B_r = \emptyset$. We are going to prove the stronger statement that the entry $D[x, \kappa]$ contains a minimum-size set $S \subseteq A(G_x)$ that is (C, π) -feasible for some $C \in \mathcal{C}|_{V(G_x)}$, and $\pi : V(G_x) \rightarrow \mathbb{Z} \cap [a, b]$ with the projection of (C, π) to B_x being κ . That is, we have to prove, that the $D[x, \kappa]$'s indeed contain a set that is (C, π) -feasible as above and that it has minimum size among those. We first focus on the (C, π) -feasibility, which we prove by induction from leaf vertices to the root. Let $x \in V(T)$ be a node, such that the feasibility property holds for all vertices y that are below x in T .

Leaf nodes. As G_x is the empty graph, the statement holds trivially.

Introduce nodes. For the keys κ , we enumerate all ordered partitions $C \in \mathcal{C}|_{B_x}$. Let $C^\uparrow \in \mathcal{C}$ be an ordered partition that projects to C in B_x . The set $D[y, \kappa']$ we chose is, by induction, $(C'|_{V(G_y)}, \pi')$ -feasible on G_y . By T -compatibility of \mathcal{C} we have that there is a $C^* \in \mathcal{C}$, with $C^*|_{V(G_y)} = C'|_{V(G_y)}$ and $C^*|_{V(G) \setminus (V(G_y) \cup B_y)} = C^\uparrow|_{V(G) \setminus (V(G_y) \cup B_y)}$. Thus, by construction, our set is $(C^*|_{V(G_x)}, \pi^*|_{V(G_x)})$ -feasible, where $\pi^*(v) = \pi(v)$ and $\pi^*(u) = \pi'(u)$ for all $u \in V(G_x) \setminus \{v\}$.

Forget nodes. As $G_x = G_y$ the statement holds trivially.

Merge nodes. Let y_1 and y_2 be the two children of x . By induction, let $D[y_i, \kappa_i]$ be $(C_i|_{V(G_{y_i})}, \pi_i)$ -feasible. By T -compatibility of \mathcal{C} we have that there is a $C^* \in \mathcal{C}$ such that $C^*|_{V(G_x)} = C_1|_{V(G_x)}$ and $C^*|_{V(G) \setminus (V(G_x) \cup B_x)} = C_2|_{V(G) \setminus (V(G_x) \cup B_x)}$. With $\pi^*(u) = \pi_1(u)$ for all $u \in V(G_{y_1})$ and $\pi^*(u) = \pi_2(u)$ otherwise, we get that our chosen set is $(C^*|_{V(G_x)}, \pi^*|_{V(G_x)})$ -feasible.

Now assume for contradiction that our set is not the minimum choice among the (C, π) -feasible ones. Then there is a node $x \in V(T)$ such that our statement holds for all nodes y in the subtree of T rooted at x but not for x (with x possibly being a leaf node and the set of other nodes being empty). In particular, we made the minimum choice for all these nodes y and all κ 's. We make a case distinction based on the type of x .

Leaf nodes. G_x is empty and thus the empty set is the right choice of $D[x, ((), \varepsilon)]$.

Introduce nodes. Assume there is a set of smaller size S^* that is (C^*, π^*) -feasible for some $C^* = (C_1^*, \dots, C_t^*) \in \mathcal{C}|_{V(G_x)}$ and $\pi^* : V(G_x) \rightarrow \mathbb{Z} \cap [a, b]$ with (C^*, π^*) 's projection to B_x being κ . Choose S^* smallest possible among all such choices. By Claim 3.16, we have that $S^* \cap A(G[B_x])$ contains exactly those arcs $a = (p, q) \in A(G[B_x])$ with $p \in C_i^*$ and $q \in C_j^*$ such that either $j < i$ or $i = j$ and $\pi^*(p) - \pi^*(q) + w(a) < 0$. In especially, $S^* \cap \delta(v)$ and $D[x, \kappa] \cap \delta(v)$ are identical by (C^*, π^*) projecting to κ . Thus, $S^* \setminus \delta(v)$ is $(C^*|_{V(G_y)}, \pi^*|_{V(G_y)})$ -feasible by $G_y - (S^* \setminus \delta(v)) \subseteq G_x - S^*$ and projects down to the same κ' as κ in B_y . But $S^* \setminus \delta(v)$ is of size smaller than $D[y, \kappa|_{B_y}]$, a contradiction to $D[y, \kappa|_{B_y}]$ containing the smallest such arc set.

Forget nodes. We have that $G_x = G_y$ and thus any candidate for $D[x, \kappa]$ is also a candidate for $D[y, \kappa^*]$ by extending the κ^* in a way that matches the candidate. By taking the minimum over the possible choices of κ^* and the claim holding for y we get that $D[x, \pi]$ is indeed such a set of minimum size.

Merge nodes. Assume there is a set of smaller size S^* that is (C^*, π^*) -feasible for some $C^* = (C_1^*, \dots, C_t^*) \in \mathcal{C}|_{V(G_x)}$ and $\pi^* : V(G_x) \rightarrow \mathbb{Z} \cap [a, b]$ with (C^*, π^*) 's projection to B_x being κ . Choose S^* smallest possible among all such choices. By Claim 3.16, we have that $S^* \cap A(G[B_x])$ contains exactly those arcs $a = (p, q) \in A(G[B_x])$ with $p \in C_i^*$ and $q \in C_j^*$ such that either $j < i$ or $i = j$ and $\pi^*(p) - \pi^*(q) + w(a) < 0$. In especially, we have that $S^* \cap A(G[B_x])$,

$D[y_1, \kappa] \cap A(G[B_x])$ and $D[y_2, \kappa] \cap A(G[B_x])$ are identical (as $B_x = B_{y_1} = B_{y_2}$). Thus, from $|S^*| < |D[x, \kappa]|$, we have that $|S^* \cap A(G_{y_i})| < |D[y_i, \kappa]|$ for some $i \in \{1, 2\}$. As $S^* \cap A(G_{y_i})$ is a candidate for $D[y_i, \kappa]$, this is a contradiction to the minimality of $D[y_i, \kappa]$.

Thus, our algorithm computes a minimum-size negative directed feedback arc set of (G, w) . For the run-time, note that for every of the $\mathcal{O}(n)$ nodes of the tree decomposition T , we make a number of computation that is bounded by the number of κ 's for the node itself and it's up to two children. Any of those computations is polynomial in the size of the arc sets of the bag. Now, the number of κ 's is bounded by the number of possible functions π and the time needed to enumerate $\mathcal{C}|_{B_x}$ for any $x \in V(T)$. The number of functions π is bounded by $(b - a + 1)^{\mathcal{O}(\text{tw}(G))} = 2^{\mathcal{O}(\text{tw}(G) \log(b-a))}$, showing the overall run-time. \square

We can deduce three of our main results from this general algorithm.

Theorem 3.17. *Let (G, w, k) be a MINFB instance with weights $w : A(G) \rightarrow \{-1, 0, 1\}$. Then in time $2^{\mathcal{O}(\text{tw}(G) \log w_-)}(n + m)$, we can decide the existence of a solution to (G, w, k) .*

Proof. By Theorem 2.3, we can construct an instance of NEGATIVE DFAS in polynomial time that has the same parameters $\text{tw} + w_-$ such that the original instance is a “yes”-instance if and only if the constructed instance is.

Let \mathcal{C} consist of the single partition $(V(G))$ of $V(G)$. Compute a nice tree decomposition T of G . Note that \mathcal{C} is T -compatible. We know that any (C, π) -feasible set S for $C \in \mathcal{C}$ is a negative directed feedback arc set with π being a feasible potential of $G - S$. By Lemma 3.11, we know that for any negative directed feedback arc set S of (G, k) , $G - S$ has a feasible potential $\pi : V(G) \rightarrow \mathbb{Z} \cap [0, w_-]$. Thus, we call Lemma 3.15 with T , \mathcal{C} , $a = 0$ and $b = w_-$ and get a negative directed feedback arc set S' of minimum size. We then check whether $|S'| \leq k$ or not, and return the corresponding answer.

For the run-time, we use Lemma 3.15 and Proposition 2.7. Regarding Lemma 3.15, observe that the sets of the form $C|_{B_x}$ for any $x \in V(T)$ are exactly the sets B_x . \square

Theorem 3.18. *Given any MINFB instance (G, w, k) with weights $w : A(G) \rightarrow \{-1, 0, 1\}$, in time $2^{\mathcal{O}(\text{tw}(G)(\log \text{tw}(G) + \log w_+))}(n + m)$ we can decide the existence of a solution to (G, w, k) .*

Proof. By Theorem 2.3, we can construct an instance of NEGATIVE DFAS in polynomial time that has the same parameters $\text{tw} + w_+$ such that the original instance is a “yes”-instance if and only if the constructed instance is.

Let \mathcal{C} consist of all ordered partitions of $V(G)$. Compute a nice tree decomposition T of G . By properties of tree decompositions we have that for any $x \in V(T)$, the sets $V(G_x)$ and $V(G) \setminus (V(G_x) \setminus B_x)$ only intersect in B_x . Thus, \mathcal{C} is T -compatible as we can recombine any two ordered partitions that match on a subset B_x .

We know that any (C, π) -feasible set S for $C = (C_1, \dots, C_t) \in \mathcal{C}$ is a negative directed feedback arc set with π being a feasible potential for every $G[C_i] - S$ and

(C_1, \dots, C_t) is a (super-)partition of the strong components of $G - S$. By Lemma 3.12, we know that for any negative directed feedback arc set S of (G, k) , the strong components of $G - S$ have a feasible potential $\pi : V(G) \rightarrow \mathbb{Z} \cap [0, w_+]$. Thus, we call Lemma 3.15 with $T, \mathcal{C}, a = 0$ and $b = w_+$ and get a negative directed feedback arc set S' of minimum size. We then check whether $|S'| \leq k$ or not and return the corresponding answer.

For the run-time, we use Lemma 3.15 and Proposition 2.7. In Lemma 3.15, the sets of the form $C|_{B_x}$ for any $x \in V(T)$ can be enumerated in time $2^{\mathcal{O}(\text{tw}(G) \log \text{tw}(G))}$ by taking every ordered partition of B_x . \square

Theorem 3.19. *Let (G, w, k) be a MINFB instance with weights $w : A(G) \rightarrow \{-1, 0, 1\}$. Then, in time $2^{\mathcal{O}(\text{td}(G)^2)}(n + m)$, we can decide the existence of a solution to (G, w, k) .*

Proof. By Theorem 2.3, we can construct an instance of NEGATIVE DFAS in polynomial time that has the same parameters td such that the original instance is a “yes”-instance if and only if the constructed instance is.

Let \mathcal{C} consist of the single partition $(V(G))$ of $V(G)$. Compute a nice tree decomposition T of G . Note that \mathcal{C} is T -compatible. We know that any (C, π) -feasible set S for $C \in \mathcal{C}$ is a negative directed feedback arc set of (G, w) with π being a feasible potential of $G - S$. By Lemma 3.13, we know that for any negative directed feedback arc set S of (G, w) , $G - S$ has a feasible potential $\pi : V(G) \rightarrow \mathbb{Z} \cap [0, 2^{\text{td}(G)}]$. Thus, we call Lemma 3.15 with $T, \mathcal{C}, a = 0$ and $b = 2^{\text{td}(G)}$ and get a negative directed feedback arc set S' of minimum size. We then check whether $|S'| \leq k$ or not and return the corresponding answer.

For the run-time, we use Lemma 3.15 and Proposition 2.7. In Lemma 3.15, the sets of the form $C|_{B_x}$ for any $x \in V(T)$ can be enumerated in constant time as they are exactly the set (B_x) . Last but not least, we have $\text{tw}(G) \leq \text{td}(G)$, and thus an overall run-time of $2^{\mathcal{O}(\text{td}(G) \log(2^{\text{td}(G)}))}(n + m) = 2^{\mathcal{O}(\text{td}(G)^2)}(n + m)$. \square

3.5 Algorithm for $\{-1, 1\}$ entries with Few Negative Arcs

In this section we give an algorithm when parameterized by w_- . However, this algorithm works only for weights of the form $w : A(G) \rightarrow \{-1, 1\}$. As stated in the results overview (see Section 1.2), the parameterized complexity of MINFB with respect to parameter is open for right-hand side $b \in \{-1, 0, 1\}$, and $\mathbf{W}[1]$ -hard for general integral weights. In the case of $\{-1, +1\}$ values, finding a negative directed feedback arc set however becomes easy by the following observation.

Lemma 3.20. *Let G be a directed graph with arc weights $w : A(G) \rightarrow \{-1, 1\}$. Then any negative cycle of G has length at most $2w_-$.*

Proof. Any cycle C of length more than $2w_-$ contains at least $(w_- + 1)$ -many arcs of weight $+1$ and at most w_- -many arcs of weight -1 , therefore its total weight is non-negative. \square

Combining Lemma 3.20 and Lemma 3.3 leads to the following theorem.

Theorem 3.21. *There is an algorithm solving MINFB with right-hand sides in $\{-1, 1\}$ in time $\mathcal{O}((2w_-)^k n^2 m)$.*

Proof. By Theorem 2.3, we can construct an instance of NEGATIVE DFAS in polynomial time that has the same parameters $k + w_-$ such that the original instance is a “yes”-instance if and only if the constructed instance is. The theorem follows by combining Lemma 3.3 and Lemma 3.20. \square

Note that the run-time can be improved to $\mathcal{O}((k + 1)w_-^k n^2 m)$ by guessing the number of -1 arcs in our solution first and enumerating all sets of that size. Then the algorithm of Lemma 3.3 can be modified to only make subroutine calls on arcs of weight $+1$, leading to an improved run-time. Moreover, note that we can assume $k \leq w_-$ by Theorem 2.4, which shows that NEGATIVE DFAS with arc weights $w : A(G) \rightarrow \{-1, 1\}$ is fixed-parameter tractable in w_- .

3.6 Algorithm for $\{-1, 1\}$ Weights with Few Positive Arcs

We will now study weight functions $w : A(G) \rightarrow \{-1, +1\}$ when parameterized by $k + w_+$. As stated in the results overview in Section 1.2 this parameter is open for weights of the form $w : A(G) \rightarrow \{-1, 0, 1\}$ and W[1]-hard for general integral weights.

The main observation for our algorithm is made in the following lemma:

Lemma 3.22. *Let G be a directed graph with arc weights $w : A(G) \rightarrow \{\pm 1\}$. Then either G has a negative cycle of length at most $2(w_+)^2 + 2w_+$, or every negative cycle C has some arc $a \in A(C)$ that lies only on negative cycles of G .*

Proof. First, if G has no negative cycles then we are done. We are also done if there is a negative cycle of length at most $2(w_+)^2 + 2w_+$. So, assume that G has a negative cycle and all negative cycles have length at least $2(w_+)^2 + 2w_+ + 1$.

Suppose, for sake of contradiction, that each arc a in a negative cycle lies on a cycle C_a of non-negative weight. Let C be a shortest negative cycle in G . By the argumentation above, C has length at least $2(w_+)^2 + 2w_+ + 1$. In particular, C has length at least $w_+(w_+ + 1) + 1$ and contains at most w_+ arcs of weight $+1$; all other arcs of C have weight -1 . By pigeonhole principle, there must be a segment P of C consisting of $w_+ + 1$ arcs of weight -1 .

By assumption, for every arc $a = (v, w) \in A(P)$ we have a $w \rightarrow v$ -path R_a of length at most $|C_a| - 1$. As $w(C_a) \geq 0$ and the C_a 's contain at most w_+ arcs of weight $+1$ and all other arcs have weight -1 the cycles have length at most $2w_+$. Thus, the reverse paths R_a have length $|R_a| \leq |C_a| - 1 \leq 2w_+ - 1$. Say P is an $s \rightarrow t$ -path, then by concatenating the R_a 's with $a \in P$ we get an $t \rightarrow s$ -walk R' which contains a $t \rightarrow s$ -path R . The path R contains at most w_+ arcs of weight $+1$, thus $w(R) \leq w_+$.

Consider the closed walk $O = P \circ R$. Then $w(O) = w(P) + w(R) = -(w_+ + 1) +$

$w(R) \leq -1$. Thus, O contains a negative cycle C' . Eventually,

$$\begin{aligned} |\ell(C')| &\leq |O| \\ &= |P| + |R| \\ &\leq |P| + \sum_{e \in E(P)} |R_e| \\ &\leq 2w_+(w_+ + 1) \\ &< 2(w_+)^2 + 2w_+ + 1 \leq |\ell(C)| \end{aligned}$$

yields a contradiction to the fact that C was a shortest negative cycle. \square

This lemma forms the basis of our algorithm, Algorithm 1. First, the algorithm checks for negative cycles with up to $2(w_+)^2 + 2w_+$ arcs. It then guesses the arc contained in a solution like the algorithm alluded to in Lemma 3.3. Afterwards, we are left with a digraph without short negative cycles. We now identify the set U of arcs which are not part of a non-negative cycle. Then, we know that $G - S$ may not contain a cycle on which an arc of U lies, as this cycle would be negative by definition of U . Likewise, any negative cycle in G has some arc in U by the previous lemma. For general sets U , this is exactly the DIRECTED SUBSET FEEDBACK ARC SET problem.

DIRECTED SUBSET FEEDBACK ARC SET

Input: A graph G , an arc set $U \subseteq A(G)$ and an integer $k \in \mathbb{Z}_{\geq 0}$.

Task: Find an arc set $X \subseteq A(G)$ of size at most k such that every cycle of $G - X$ is disjoint from U or decide that no such set exists.

The DIRECTED SUBSET FEEDBACK ARC SET problem was shown to be fixed-parameter tractable for parameter k by Chitnis et al. [17].

Proposition 3.23 ([17]). DIRECTED SUBSET FEEDBACK ARC SET is solvable in time $2^{\mathcal{O}(k^3)} \cdot n^{\mathcal{O}(1)}$.

Before we prove correctness and run-time, we show how to detect the set U of all arcs which lie only on negative cycles. We first argue that this problem is NP-hard even for weights $w : A(G) \rightarrow \{-1, +1\}$. To this end, we provide a reduction from the HAMILTONIAN s - t -PATH problem, which for a directed graph H and vertices $s, t \in V(H)$ asks for an $s \rightarrow t$ -path in H visiting each vertex of H exactly once. Its NP-hardness was shown by Karp [34]. The reduction works as follows: Take the original directed graph H and two vertices $s, t \in V(H)$ which we want to test for the existence of a Hamiltonian path starting in s and ending in t . Add a path P of length $n - 1$ from t to s to the graph. Assign weight $+1$ to each arc of H , and weight -1 to each arc of P . Then an arc of P lies on a cycle of non-negative length if and only if there is a Hamiltonian $s \rightarrow t$ -path in H .

However, for this construction of weights w we have $w_+ \in \Omega(n)$. We will now show that the task is indeed fixed-parameter tractable when parameterized by w_+ . For

Algorithm 1: NegativeCycleDeletion

Input : A directed graph G with arc weights $w : A(G) \rightarrow \mathbb{Z}$ and $k \in \mathbb{Z}_{\geq 0}$.
Output: A set $S \subseteq A(G)$ of at most k arcs such that $G - S$ has no negative cycle, or **false** if no such set exists.

- 1 **if** $k < 0$ **then**
- 2 | **return false.**
- 3 **if** *there is some negative cycle C of length at most $2(w_+)^2 + 2w_+$ in G* **then**
- 4 | Branch on deleting an arc of C and try to solve with $k - 1$.
- 5 **else**
- 6 | Identify the set U of all arcs which do not lie on a non-negative cycle.
- 7 | **return DirectedSubsetFeedbackArcSet** (G, U, k).

that, the main observation is that every non-negative cycle has length at most $2w_+$. We now consider the WEIGHTED LONGEST PATH problem: given a directed graph G with arc weights $w : A(G) \rightarrow \mathbb{R}$ and numbers $W \in \mathbb{R}, \ell \in \mathbb{Z}_{\geq 0}$, the task is to find a path of length exactly ℓ and weight at least W in G . Zehavi [48] gave a fast algorithm for WEIGHTED LONGEST PATH, based on color coding-related techniques and representative sets.

Proposition 3.24 ([48]). WEIGHTED LONGEST PATH can be solved in time $2^{\mathcal{O}(\ell)} \cdot \mathcal{O}(m \log n)$.

So given an arc $a = (s, t)$, one can enumerate all path sizes ℓ from 1 to $2w_+ - 1$ and ask whether there is a $t \rightarrow s$ -path of length ℓ of weight at least $-w(a)$. This way one can detect a non-negative cycle containing a .

Corollary 3.25. Let G be a directed graph, let $w : A(G) \rightarrow \{\pm 1\}$ be a function and $(s, t) \in A(G)$ be an arc. Then one can detect in time $2^{\mathcal{O}(w_+)} \cdot m \log n$ if $a = (s, t)$ is part of some non-negative cycle C .

Finally, we argue the correctness and run-time of Algorithm 1, proving the following:

Theorem 3.26. Algorithm 1 is correct and solves an instance of MINFB with right-hand sides in $\{-1, 1\}$ in time $2^{\mathcal{O}(k^3 + w_+ + k \log w_+)} \cdot n^{\mathcal{O}(1)}$.

Proof. By Theorem 2.3, we can construct an instance of NEGATIVE DFAS in polynomial time that has the same parameters $k + w_+$ such that the original instance is a “yes”-instance if and only if the constructed instance is.

We detect with help of Proposition 3.1 in time $\mathcal{O}(n^2 m)$ whether there is a negative cycle in G , and if the minimum-length negative cycle C has length at most $2(w_+)^2 + 2(w_+)$. If such a cycle C exists, we split into $|\ell(C)|$ instances, one for each $a \in A(C)$, calling our algorithm recursively with parameter k decreased by one on $G - a$. This is correct, as one of the arcs of C needs to be deleted, and we just search for every arc if there is a solution containing this arc.

Otherwise, all negative cycles have length at least $2(w_-)^2 + 2(w_-) + 1$, and by Lemma 3.22 every negative cycle must have an arc not contained in a non-negative

cycle. Choose one arc of each such cycle and gather them in the set $U_{<0}$. By definition of the set U , in our algorithm we have $U_{<0} \subseteq U$. SUBSET DFAS for (G, W, k) now asks for a set S of size at most k such that $G - S$ has no cycle containing an arc of W . As $U_{<0} \subseteq U$, we get that a solution for (G, U, k) is also a solution for $(G, U_{<0}, k)$. We now show that the reverse direction also holds, thus solving the original problem. For this, let $S_{<0}$ be a solution for $(G, U_{<0}, k)$ and C be a cycle in $G - S_{<0}$. If there is no such cycle, we are done. Otherwise, we know that C cannot be a cycle of negative weight as every cycle of negative weight has an arc in $U_{<0}$. But as $w(C) \geq 0$, our cycle C cannot contain an arc of U as those are not contained in non-negative cycles. Thus, $S_{<0}$ is a solution for (G, U, k) . Also, all cycles in $G - S_{<0}$ are non-negative, and therefore $S_{<0}$ is also a solution to our NEGATIVE DFAS instance (G, k) .

This shows the correctness of Algorithm 1. The run-time can be bounded as follows. Detecting cycles in line 4 can be done in time $\mathcal{O}((2(w_+)^2 + 2w_+)nm)$. The branching step then creates up to $2(w_+)^2 + 2w_+$ instances with the parameter k decreased by one. As there is no other recursive call to this algorithm, we have at most $(2(w_+)^2 + 2w_+ + 1)^k$ instances for which this algorithm is called. In the end, we call the algorithm alluded to in Corollary 3.25 to compute the set U which takes time $2^{\mathcal{O}(w_+)} \cdot m^2 \log n$, as it is called for every arc. By Proposition 3.23, the final call to the DIRECTED SUBSET FEEDBACK ARC SET oracle takes time $2^{\mathcal{O}(k^3)} \cdot n^{\mathcal{O}(1)}$.

Thus, we obtain an overall run-time of

$$(2w_+^2 + 2w_+ + 1)^k \cdot \left[(w_+^2 + 2w_+) \cdot \mathcal{O}(nm) + 2^{\mathcal{O}(w_+)} \cdot \mathcal{O}(m^2 \log n) + 2^{\mathcal{O}(k^3)} \cdot n^{\mathcal{O}(1)} \right],$$

which simplifies to $(w_+)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(k^3 + w_+)} \cdot n^{\mathcal{O}(1)} = 2^{\mathcal{O}(k^3 + w_+ + k \log w_+)} \cdot n^{\mathcal{O}(1)}$. \square

4 Parameterized Intractability Results

4.1 NP-Hardness for Number of Positive Arcs

For completeness of the hardness results, this section contains a short observation about the equivalence of DFAS and NEGATIVE DFAS instances where all arc weights are -1 . This implies that NEGATIVE DFAS is NP-hard even in the case where all arc weights are -1 . This implies the hardness result for MINFB with parameter w_+ and arc weights $w : A(G) \rightarrow \{-1, 0, 1\}$.

Theorem 4.1. *NEGATIVE DFAS is NP-hard, even if all arc weights are -1 .*

Proof. We show the theorem by a reduction from DFAS, which is NP-hard. Let (G, k) be an instance of DFAS. We claim that (G, w, k) with $w \equiv -1$ is an equivalent instance of NEGATIVE DFAS. Indeed, for every $S \subseteq A(G)$ there are no cycles in $G - S$ if and only if $G - S$ with weights -1 contains no negative cycles. \square

4.2 NP-Hardness for Constant Pathwidth

In this section we show that MINFB is NP-hard even for constraint matrices A whose pathwidth is bounded by 6. To this end, we reduce PARTITION to NEGATIVE DFAS in directed graphs whose underlying undirected graph has pathwidth at most 6.

PARTITION

Input: A set $\mathcal{A} = \{a_1, \dots, a_n\}$ of positive integers.

Task: Find a subset \mathcal{A}' such that $\sum_{a_i \in \mathcal{A}'} a_i = \sum_{a_i \in \mathcal{A} \setminus \mathcal{A}'} a_i$ or decide that no such subset exists.

Using $A = \sum_{i=1}^n a_i$, we can reformulate PARTITION as the problem of finding a subset \mathcal{A}' such that \mathcal{A}' and $\mathcal{A} \setminus \mathcal{A}'$ both sum up to $\frac{A}{2}$ (or no such subset exists). Karp [34] showed that PARTITION is NP-complete.

Proposition 4.2 (Karp [34]). *PARTITION is NP-complete.*

Now we are ready to state our hardness result.

Theorem 4.3. *NEGATIVE DFAS is NP-hard, even for graphs of pathwidth 6 and one arc of positive weight.*

Proof. Let \mathcal{A} be an instance of PARTITION and $A = \sum_{a_i \in \mathcal{A}} a_i$. For every number $a_i \in \mathcal{A}$ we construct a gadget G_i as follows (see Figure 1 for an illustration). Let $V^{(i)} = \{s_i^{(j)}, t_i^{(j)}, x_i^{(j)}, y_i^{(j)} \mid j = 1, 2\}$ be the set of vertices in G_i . There are three different types of arcs. The first arc set $A_1^i = \{(x_i^{(j)}, y_i^{(j)}) \mid j = 1, 2\}$ with arc weight $-a_i$, contains the arcs we will consider for deletion later. The second arc set $A_2^i = \{(y_i^{(j)}, x_i^{(j+1)}), (y_i^{(j)}, x_i^{(j-1)}) \mid j = 1, 2\}$ with arc weight 0, contains arcs which enforce the deletion of arcs from the first arc set by inducing negative cycles. The last arc set $A_3^i = \{(s_i^{(j)}, t_i^{(j)}), (s_i^{(j)}, x_i^{(j)}), (y_i^{(j)}, t_i^{(j)}) \mid j = 1, 2\}$ with arc weight 0, contains arcs that connect the vertices $s_i^{(j)}$ and $t_i^{(j)}$ to the rest of the gadget.

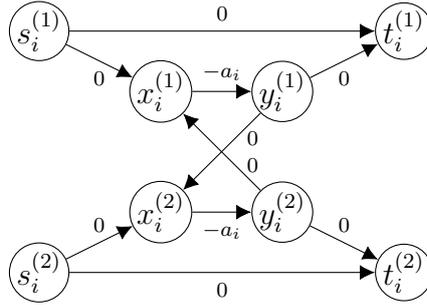


Figure 1: The gadget graph G_i .

The whole gadget G_i is then defined as $(V^{(i)}, A_1^{(i)} \cup A_2^{(i)} \cup A_3^{(i)})$. Out of these gadgets we construct the graph (G, w) of our NEGATIVE DFAS instance by taking the union of all gadgets G_i for $1 \leq i \leq n$, where we identify $t_i^{(j)} = s_{i+1}^{(j)}$ for $j = 1, 2$ and $i \in \{1, \dots, n-1\}$. Additionally, we add two vertices s and t with the arcs $(s, s_1^{(j)})$ and $(t_n^{(j)}, t)$ for $j = 1, 2$ of weight 0. Finally, we add the arc (t, s) of weight $\frac{A}{2}$. We then choose (G, w, k) with $k = n$ as our NEGATIVE DFAS instance.

Next, we show that \mathcal{A} has a solution if and only if (G, w, n) has one.

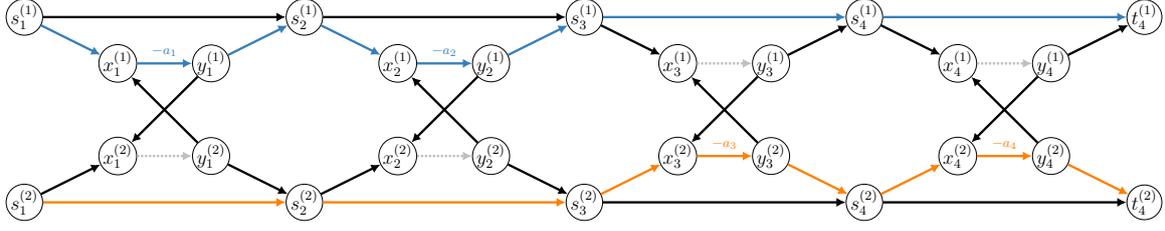


Figure 2: Union of four gadgets after the deletion of negative cycles. Deleted arcs are shown with a dotted gray line. The blue path shows the shortest path from $s_1^{(1)}$ to $t_4^{(1)}$, and the orange path shows the shortest path from $s_1^{(2)}$ to $t_4^{(2)}$.

Claim 4.4. *If \mathcal{A} has a solution \mathcal{A}' , then there is a solution S^* to (G, w, n) .*

Proof of Claim 4.4. For every $i \in \{1, \dots, n\}$, we define an $s_i \in S^*$ by

$$s_i = \begin{cases} (x_i^{(1)}, y_i^{(1)}), & \text{if } a_i \in \mathcal{A}', \\ (x_i^{(2)}, y_i^{(2)}), & \text{if } a_i \notin \mathcal{A}' . \end{cases}$$

Suppose, for sake of contradiction, that $G - S^*$ contains a negative cycle C . Observe that for each $i \in \{1, \dots, n\}$ the graph $G_i - s_i$ contains no negative cycle. Moreover, any gadget G_i can only be entered through one $s_i^{(j)}$ and left through $t_i^{(j)} = s_{i+1}^{(j)}$ with the same j . Thus, C has to start in s , pass through the gadgets G_i in order either on the $j = 1$ or the $j = 2$ side, then go to t and eventually use the backward arc (t, s) . As the backward arc has weight $\frac{A}{2}$, C contains an $s \rightarrow t$ -path P of weight less than $\frac{A}{2}$. Furthermore, P uses either only vertices with index $j = 1$ or $j = 2$. The only arcs of negative weight with index $j = 1$ in $G - S^*$ are those $(x_i^{(1)}, y_i^{(1)})$ with $a_i \notin \mathcal{A}'$. For these we have $\sum_{a_i \notin \mathcal{A}'} w((x_i^{(1)}, y_i^{(1)})) = \sum_{a_i \notin \mathcal{A}'} -a_i = -\frac{A}{2}$, as \mathcal{A}' is a solution to \mathcal{A} . Likewise, the only arcs of negative weight with index $j = 2$ in $G - S^*$ are those $(x_i^{(2)}, y_i^{(2)})$ with $a_i \in \mathcal{A}'$. For these we have $\sum_{a_i \in \mathcal{A}'} w((x_i^{(2)}, y_i^{(2)})) = -\sum_{a_i \in \mathcal{A}'} a_i = -\frac{A}{2}$, as \mathcal{A}' is a solution to \mathcal{A} . Thus, in any case, P has weight at least $-\frac{A}{2}$, a contradiction. \blacksquare

Claim 4.5. *If (G, w, n) has a solution S^* then \mathcal{A} has a solution \mathcal{A}' .*

Proof of Claim 4.5. We choose $\mathcal{A}' = \{a_i \mid (x_i^{(1)}, y_i^{(1)}) \in S^*\}$. Suppose, for sake of contradiction, that \mathcal{A}' is not a solution. Then we have that either $\sum_{a_i \in \mathcal{A}'} a_i > \frac{A}{2}$ or $\sum_{a_i \notin \mathcal{A}'} a_i > \frac{A}{2}$. We consider now a minimum-weight $s_1^{(j)} \rightarrow t_n^{(j)}$ -path P_j in $G - S^*$ for $j = 1, 2$ as depicted in Figure 2. Note that for every gadget G_i , our solution S^* has to contain one of the arcs $(x_i^{(1)}, y_i^{(1)})$, $(x_i^{(2)}, y_i^{(2)})$, $(y_i^{(1)}, x_i^{(2)})$ and $(y_i^{(2)}, x_i^{(1)})$. As S^* contains only n elements (one for each gadget), the arcs $(s_i^{(j)}, t_i^{(j)})$ are always available and have weight 0. Moreover, if $(x_i^{(j)}, y_i^{(j)})$ is undeleted, the $s_i^{(j)} \rightarrow t_i^{(j)}$ -path $(s_i^{(j)}, x_i^{(j)}) \circ (x_i^{(j)}, y_i^{(j)}) \circ (y_i^{(j)}, t_i^{(j)})$ of weight $-a_i$ exists. Thus, a minimum-weight $s_1^{(j)} \rightarrow t_n^{(j)}$ -path P_j has weight at most $-\sum_{a_i \in \mathcal{A}_j} a_i$ where \mathcal{A}_j is the set of a_i 's for which $(x_i^{(j)}, y_i^{(j)})$ is undeleted. By choice of \mathcal{A}' , we have that $\mathcal{A}_1 = \mathcal{A} \setminus \mathcal{A}'$ and $\mathcal{A}_2 \supseteq \mathcal{A}'$. So, if $\sum_{a_i \in \mathcal{A}'} a_i > \frac{A}{2}$, we have that $w(P_2) \leq -\sum_{a_i \in \mathcal{A}_2} a_i \leq -\sum_{a_i \in \mathcal{A}'} a_i < -\frac{A}{2}$. If instead

$\sum_{a_i \notin \mathcal{A}'} a_i > \frac{A}{2}$, we have that $w(P_1) \leq -\sum_{a_i \in \mathcal{A}_1} a_i \leq -\sum_{a_i \notin \mathcal{A}'} a_i < -\frac{A}{2}$. Thus, in any case we have an $s_1^{(j)} \rightarrow t_n^{(j)}$ -path P_j in $G - S^*$ of weight less than $-\frac{A}{2}$ for some $j \in \{1, 2\}$.

Again, our solution S^* contains exactly one arc of every gadget and thus no arc incident to s or t . So, we can complete this path P_j to a cycle $(s, s_1^{(j)}) \circ P_j \circ (t_n^{(j)}, t) \circ (t, s)$ in $G - S^*$ of weight $w(P_j) + \frac{A}{2} < 0$, a contradiction to $G - S^*$ containing no negative cycles. ■

This completes the reduction from PARTITION to NEGATIVE DFAS and therefore shows the NP-hardness of NEGATIVE DFAS. It only remains to bound the pathwidth of the generated instances.

We show that the underlying graph of G has pathwidth at most 6, by providing a path decomposition (P, \mathcal{B}) of G of width 6. Let P be the path on $2n + 1$ vertices, corresponding to bags $B_1, \dots, B_{2n+1} \in \mathcal{B}$:

- $B_{2i-1} = \{s, s_i^{(1)}, s_i^{(2)}, x_i^{(1)}, x_i^{(2)}, y_i^{(1)}, y_i^{(2)}\}$ for $i = 1, \dots, n$
- $B_{2i} = \{s, s_i^{(1)}, s_i^{(2)}, y_i^{(1)}, y_i^{(2)}, t_i^{(1)}, t_i^{(2)}\}$ for $i = 1, \dots, n$
- $B_{2n+1} = \{s, t_n^{(1)}, t_n^{(2)}, t\}$

Also, the arc (t, s) is the only arc of positive weight. □

4.3 W[1]-hardness for Treedepth and Few Positive Arcs

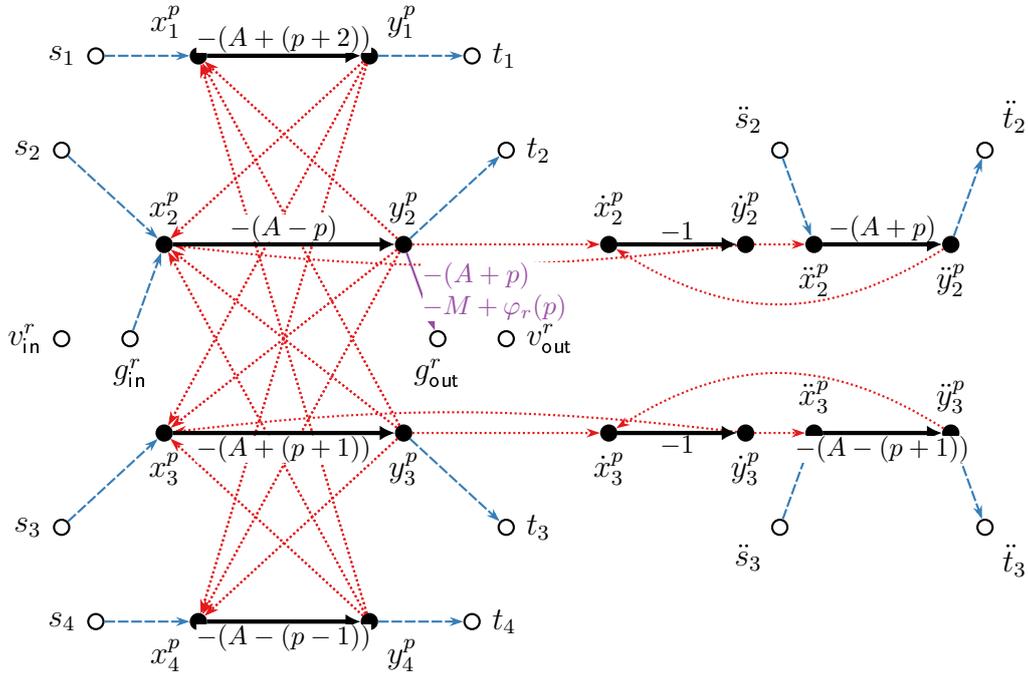
In this section we prove W[1]-hardness for MINFB when parameterized by treedepth and number of positive arcs, by reducing NEGATIVE DFAS to CLIQUE.

Theorem 4.6. *NEGATIVE DFAS is W[1]-hard when parameterized by the treedepth and number of positive arcs.*

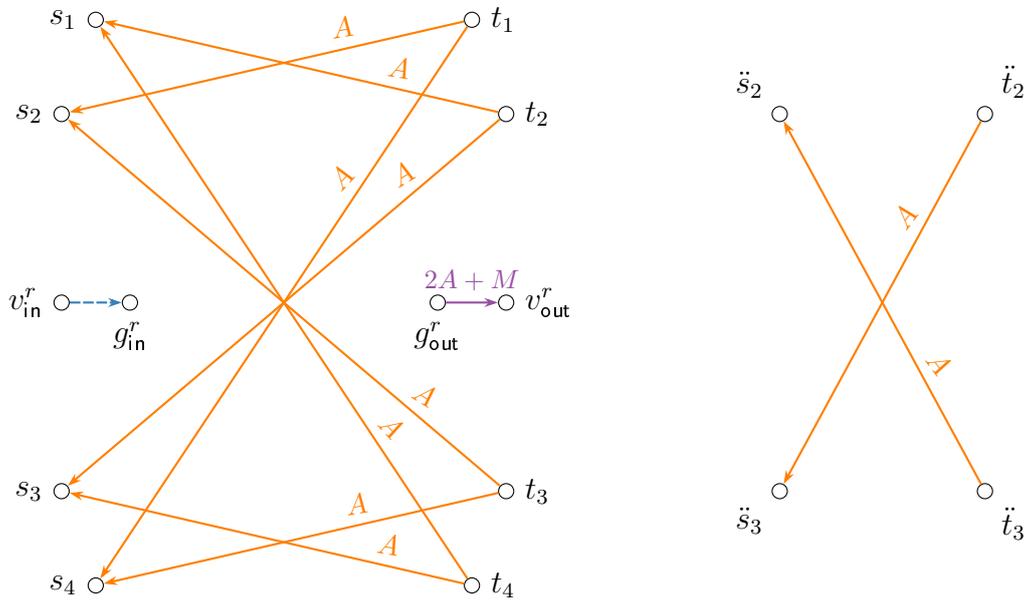
Proof. We prove the theorem by reduction from CLIQUE. The input of CLIQUE consists of an undirected graph G and an integer k . The question is, whether there is a vertex set $X \subseteq V(G)$ of at least k vertices such that $G[X]$ forms a complete graph. CLIQUE is W[1]-hard parameterized by k , see e.g. [21].

First we introduce a meta gadget that allows us to model different choices as minimum deletion sets influencing the minimum weight of a path between prescribed pairs of vertices. See Figure 3 for an illustration. Let $a, b \in \mathbb{Z}_{>0}$ be positive integers. For every $r \in \{1, \dots, b\}$, let $\varphi_r : \{1, \dots, a\} \rightarrow \mathbb{Z} \cap [-M, M]$ be some function. We define the gadget $R^{a,b}$ in the following way. Let $A = a + 4$. For every $p \in \{-1, 0, \dots, a + 1\}$ we have a smaller gadget consisting of the vertices x_i^p, y_i^p for $i \in \{1, 2, 3, 4\}$ and $\hat{x}_i^p, \hat{y}_i^p, \check{x}_i^p, \check{y}_i^p$ for $i \in \{2, 3\}$ that are interconnected by the following arcs:

- arc (x_1^p, y_1^p) of weight $-(A + (p + 2))$,
- arc (x_2^p, y_2^p) of weight $-(A - p)$,
- arc (x_3^p, y_3^p) of weight $-(A + (p + 1))$,



(a) Meta gadget (sub-level view).



(b) meta Gadget (top-level view).

Figure 3: Overview of construction of a meta gadget. The dotted red arcs have weight -1 , the dashed blue arcs have weight 0 , the solid orange arcs have weight A . The black arcs model the choice we make for each gadget and the purple arcs are used to extract information from this choice. For both of these arc types the weight is individual per arc.

- arc (x_4^p, y_4^p) of weight $-(A - (p - 1))$,
- arc $(\dot{x}_i^p, \dot{y}_i^p)$ of weight -1 for $i \in \{2, 3\}$,
- arc $(\ddot{x}_2^p, \ddot{y}_2^p)$ of weight $-(A + p)$,
- arc $(\ddot{x}_3^p, \ddot{y}_3^p)$ of weight $-(A - (p + 1))$,
- arcs (y_i^p, x_j^p) for any distinct $i, j \in \{1, 2, 3, 4\}$ all of weight -1 ,
- arcs $(y_i^p, \dot{x}_i^p), (\dot{y}_i^p, x_i^p), (\dot{y}_i^p, \ddot{x}_i^p), (\ddot{y}_i^p, \dot{x}_i^p)$ for $i \in \{2, 3\}$ all of weight -1 .

For $p = -1$, we duplicate the arc (x_1^{-1}, y_1^{-1}) and for $p = a + 1$, we duplicate the arc (x_4^{a+1}, y_4^{a+1}) . Moreover, we duplicate the arcs $(\dot{y}_i^p, \dot{x}_i^p), (\ddot{y}_i^p, \ddot{x}_i^p)$ for $i \in \{2, 3\}$.

Shared by all these mini gadgets, there are vertices s_i, t_i for $i \in \{1, 2, 3, 4\}$. These are connected to the previous vertices by the arcs (s_i, x_i^p) and (y_i^p, t_i) for $i \in \{1, 2, 3, 4\}$, all of weight 0. The following arcs run between the shared vertices: (t_i, s_j) for $i, j \in \{1, 2, 3, 4\}$ with $|i - j|$ odd, all of weight A . Moreover there are vertices \ddot{s}_i, \ddot{t}_i for $i \in \{2, 3\}$. These are adjacent to the arcs $(\ddot{s}_i, \ddot{x}_i^p)$ and $(\ddot{y}_i^p, \ddot{t}_i)$ of weight 0 for $i \in \{2, 3\}$, as well as to the arcs (\ddot{t}_2, \ddot{s}_3) and (\ddot{t}_3, \ddot{s}_2) , both of weight A .

This defines the functional part of the gadget. To extract the information we want, we add for every $r \in \{1, \dots, b\}$ the vertices g_{in}^r and g_{out}^r to gather the information of the mini gadgets. For every $p \in \{1, \dots, a\}$ these vertices are connected to the mini-gadgets by an arc (g_{in}^r, x_2^p) of weight 0 and an arc $(y_2^p, g_{\text{out}}^r)$ of weight $-(A + p) - M + \varphi_r(p)$.

Finally, for every $r \in \{1, \dots, b\}$, the gadget $R^{a,b}$ contains the vertices v_{in}^r and v_{out}^r that defines the interface to the outside. These are connected to g_{in}^r and g_{out}^r by an arc $(v_{\text{in}}^r, g_{\text{in}}^r)$ of weight 0 and an arc $(g_{\text{out}}^r, v_{\text{out}}^r)$ of weight $2A + M$.

We prove a series of claims about these meta-gadgets.

Claim 4.7. *Any solution to $R^{a,b}$ as NEGATIVE DFAS instance has size at least $5(a + 3)$. Moreover, any solution of size at most $5(a + 3)$ deletes*

- exactly three of the four arcs (x_i^p, y_i^p) with $i \in \{1, 2, 3, 4\}$,
- exactly one of the two arcs $(\dot{x}_2^p, \dot{y}_2^p)$ and $(\dot{x}_3^p, \dot{y}_3^p)$, and
- exactly one of the two arcs $(\ddot{x}_2^p, \ddot{y}_2^p)$ and $(\ddot{x}_3^p, \ddot{y}_3^p)$.

Proof of Claim 4.7. We prove that any solution to $R^{a,b}$ as NEGATIVE DFAS instance has to delete at least five arcs from every mini-gadget. For any $p \in \{-1, 0, \dots, a + 1\}$ and consider the following cycles. For distinct $i, j \in \{1, 2, 3, 4\}$ there is a cycle

$$(x_i^p, y_i^p) \circ (y_i^p, x_j^p) \circ (x_j^p, y_j^p) \circ (y_j^p, x_i^p),$$

which has negative weight as all arcs are negative. Moreover, there are the cycles

$$(\ddot{x}_i^p, \ddot{y}_i^p) \circ (\ddot{y}_i^p, \dot{x}_i^p) \circ (\dot{x}_i^p, \dot{y}_i^p) \circ (\dot{y}_i^p, \ddot{x}_i^p)$$

for $i \in \{2, 3\}$, both of which are negative as all their arcs are negative. Now, any set of four arcs is disjoint from at least one of these cycles. So every solution deletes at

least five arcs from every mini-gadget. As there are $a + 3$ arc-disjoint mini-gadgets, every solution has size at least $5(a + 3)$.

Note, any solution that deletes exactly five arcs from a mini-gadget must use at least two of these arcs to hit cycles involving the dotted vertices. Thus, there are only three arcs to intersect the cycles on non-dotted vertices. As the cycles for (i, j) and (j, i) are distinct (but not disjoint), these three arcs have the form (x_i^p, y_i^p) as otherwise one of the non-dotted cycles is not hit. To hit the two cycles involving dotted vertices, a solution has to spend exactly one arc on every cycle. As the arcs $(\dot{y}_i^p, \ddot{x}_i^p)$ and $(\dot{y}_i^p, \dot{x}_i^p)$ are doubled, the solution must delete either $(\dot{x}_i^p, \dot{y}_i^p)$ or $(\ddot{x}_i^p, \dot{y}_i^p)$ for $i \in \{2, 3\}$. ■

Claim 4.8. *Consider $R^{a,b}$ as NEGATIVE DFAS instance. Then for every $p^* \in \{1, \dots, a\}$ there is a solution S_{p^*} of size $5(a + 3)$ such any $v_{in}^r \rightarrow v_{out}^r$ -path in $R^{a,b} - S_{p^*}$ has weight $\varphi^r(p^*)$ for any $r \in \{1, \dots, b\}$.*

Proof of Claim 4.8. Let S_{p^*} consist of the following arcs with non-dotted endpoints:

- for $-1 \leq p < p^* - 1$, the arcs (x_2^p, y_2^p) , (x_3^p, y_3^p) , and (x_4^p, y_4^p) ,
- for $p^* - 1$ the arcs $(x_1^{p^*-1}, y_1^{p^*-1})$, $(x_2^{p^*-1}, y_2^{p^*-1})$ and $(x_4^{p^*-1}, y_4^{p^*-1})$,
- for p^* the arcs $(x_1^{p^*}, y_1^{p^*})$, $(x_3^{p^*}, y_3^{p^*})$ and $(x_4^{p^*}, y_4^{p^*})$,
- for $p^* < p \leq a + 1$, the arcs (x_1^p, y_1^p) , (x_2^p, y_2^p) , and (x_3^p, y_3^p) .

Additionally, S_{p^*} contains the following arcs with dotted endpoints:

- for $-1 \leq p < p^* - 1$, the arcs $(\ddot{x}_2^p, \dot{y}_2^p)$ and $(\ddot{x}_3^p, \dot{y}_3^p)$,
- for $p^* - 1$ the arcs $(\ddot{x}_2^{p^*-1}, \dot{y}_2^{p^*-1})$ and $(\dot{x}_3^{p^*-1}, \dot{y}_3^{p^*-1})$,
- for p^* the arcs $(\dot{x}_2^{p^*}, \dot{y}_2^{p^*})$ and $(\ddot{x}_3^{p^*}, \dot{y}_3^{p^*})$,
- for $p^* < p \leq a + 1$, the arcs $(\dot{x}_2^p, \dot{y}_2^p)$ and $(\ddot{x}_3^p, \dot{y}_3^p)$.

We have to check that $R^{a,b} - S_{p^*}$ has no cycle of negative weight. Note that the vertices x_i^p where (x_i^p, y_i^p) is deleted have out-degree zero and thus are not part of any cycles. The same holds for the vertices y_i^p where (x_i^p, y_i^p) is deleted as they have in-degree zero. The argument also holds for \ddot{x}_i^p and \dot{y}_i^p with $(\ddot{x}_i^p, \dot{y}_i^p)$ deleted. In addition, \dot{x}_i^p and \dot{y}_i^p are never part of any cycle, as for any p either $(\dot{x}_i^p, \dot{y}_i^p)$ is deleted, and they have either in- or out-degree zero, or (x_i^p, y_i^p) and $(\ddot{x}_i^p, \dot{y}_i^p)$ are deleted, of which at least one is part of any cycle involving \dot{x}_i^p and \dot{y}_i^p . Moreover, the vertices $v_{in}^r, v_{out}^r, g_{in}^r$ and g_{out}^r are never part of any cycle in $R^{a,b}$. So the only vertices on a cycle are

- the vertices s_i, t_i (non-dotted, top level),
- the vertices x_i^p, y_i^p where (x_i^p, y_i^p) is not deleted, (non-dotted, mini-gadget),
- the vertices \ddot{s}_i, \ddot{t}_i (dotted, top level),

- the vertices $\ddot{x}_i^p, \ddot{y}_i^p$ where $(\ddot{x}_i^p, \ddot{y}_i^p)$ is not deleted, (dotted, mini-gadget).

Note that as \ddot{x}_i^p and \ddot{y}_i^p are never part of any cycle, there is no cycle involving both the non-dotted and the dotted part of our gadget. Thus, we can analyze them separately.

First we consider the non-dotted part. By leaving out the vertices that do not form a cycle (see above), the mini-gadgets form an $s_i \rightarrow t_i$ -path for certain i 's. The only remaining arcs (those that are not part of these paths) are the backwards arcs (t_i, s_j) on the top-level. We analyze the $s_i \rightarrow t_i$ -paths first. As we are interested only whether there are negative cycles and not their exact weight, we may restrict ourselves to minimum weight paths of this kind. Observe that in $R^{a,b} - S_{p^*}$,

- any $s_1 \rightarrow t_1$ -path has weight at least $-(A + p^*)$,
- the only $s_2 \rightarrow t_2$ path has weight $-(A - p^*)$,
- the only $s_3 \rightarrow t_3$ path has weight $-(A + p^*)$, and
- any $s_4 \rightarrow t_4$ -path has weight at least $-(A + p^*)$.

The backward arcs (t_i, s_j) have weight A and exist only for $|i - j|$ odd. Therefore, on every cycle we have that $s_i \rightarrow t_i$ -paths of weight at least $-(A - p^*)$ and of weight at least $-(A + p^*)$ alternate. Also, between them lies an arc of weight A . To return to the starting point a walk has to consist of only such pairs (as the backward arcs only exist for $|i - j|$ odd. Each such pair has weight at least $-(A - p^*) + A - (A + p^*) + A = 0$ and thus every closed walk in the non-dotted part of $R^{a,b} - S_{p^*}$ has non-negative weight.

For the dotted part of the gadget, note that of the arcs of the form $(\ddot{x}_i^p, \ddot{y}_i^p)$, only $(\ddot{x}_2^{p^*}, \ddot{y}_2^{p^*})$ and $(\ddot{x}_3^{p^*-1}, \ddot{y}_3^{p^*-1})$ are undeleted. So the only cycle in the dotted part is $\ddot{s}_2, \ddot{x}_2^{p^*}, \ddot{y}_2^{p^*}, \ddot{t}_2, \ddot{s}_3, \ddot{x}_3^{p^*-1}, \ddot{y}_3^{p^*-1}, \ddot{t}_3, \ddot{s}_2$ of weight $-(A + p^*) + A - (A - (p^* - 1 + 1)) + A = 0$. Thus, there are no negative cycles in $R^{a,b} - S_{p^*}$ and S_{p^*} is a solution of size $5(a + 3)$.

It remains to show that any $v_{in}^r \rightarrow v_{out}^r$ -path has weight $\varphi^r(p^*)$ for any $r \in \{1, \dots, b\}$. Note that any $v_{in}^r \rightarrow v_{out}^r$ -path in $R^{a,b}$ has to start with a subpath $v_{in}^r, g_{in}^r, x_2^p, y_2^p$ and to end with a (potentially overlapping) subpath $x_2^{p'}, y_2^{p'}, g_{out}^r, v_{out}^r$ for some (not necessarily distinct) $p, p' \in \{-1, 0, \dots, a + 1\}$. As (x_2^p, y_2^p) is contained in S_{p^*} for any $p \neq p^*$, we have that the only $v_{in}^r \rightarrow v_{out}^r$ -path is $v_{in}^r, g_{in}^r, x_2^p, y_2^p, g_{out}^r, v_{out}^r$. Its weight is

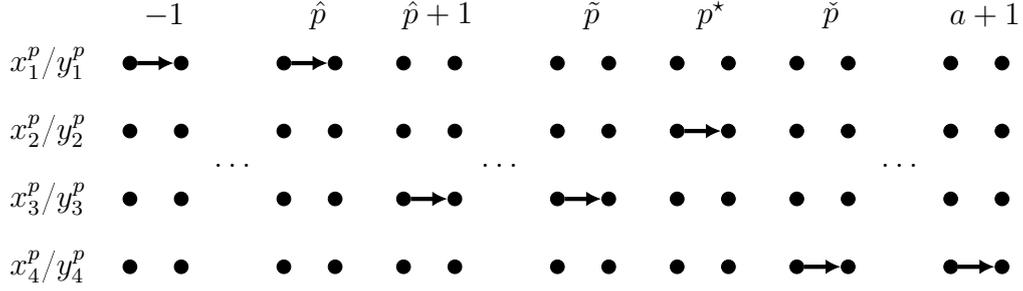
$$-(A - p^*) - (A + p^*) - M + \varphi_r(p^*) + 2A + M = \varphi_r(p^*) .$$

■

Claim 4.9. *Let S be a solution to $R^{a,b}$ as NEGATIVE DFAS instance of size $5(a + 3)$. Then there is a $p^* \in \{1, \dots, a\}$ such that any $v_{in}^r \rightarrow v_{out}^r$ -path in $R^{a,b} - S$ has weight $\varphi^r(p^*)$ for any $r \in \{1, \dots, b\}$.*

Proof of Claim 4.9. We want to prove that S has the structure of undeleted arcs that is show in Figure 4.

By Claim 4.7, our solution S has to delete exactly three arcs of type (s_i^p, t_i^p) in every mini-gadget, which is equivalent to leaving exactly one arc $(x_{i_p}^p, y_{i_p}^p)$ undeleted

Figure 4: Structure of undeleted arcs (x_i^p, y_i^p) as proven in Claim 4.9.

for every $p \in \{0, 1, \dots, a+1\}$ with $i_p \in \{1, 2, 3, 4\}$. Note that by doubling (x_1^{-1}, y_1^{-1}) and (x_4^{a+1}, y_4^{a+1}) , these are the undeleted arcs for $p \in \{-1, a+1\}$.

For ease of notation, let $w_i(p)$ denote the weight of the arc (x_i^p, y_i^p) for any $i \in \{1, 2, 3, 4\}$ and any $p \in \{-1, 0, \dots, a+1\}$. Then for any two distinct indices $p, p' \in \{-1, 0, \dots, a+1\}$ with $|i_p - i_{p'}|$ odd, we have that the cycle

$$(s_{i_p}, x_{i_p}^p) \circ (x_{i_p}^p, y_{i_p}^p) \circ (y_{i_p}^p, t_{i_p}) \circ (t_{i_p}, s_{i_{p'}}) \circ (s_{i_{p'}}, x_{i_{p'}}^{p'}) \circ (x_{i_{p'}}^{p'}, y_{i_{p'}}^{p'}) \circ (y_{i_{p'}}^{p'}, t_{i_{p'}}) \circ (t_{i_{p'}}, s_{i_p})$$

exists in $R^{a,b} - S$. This cycle we denote by $C_{p,p'}$. The weight of this cycle is $2A + w_{i_p}(p) + w_{i_{p'}}(p')$. As S is a solution to $R^{a,b}$ as NEGATIVE DFAS instance, these cycles always have non-negative weight.

Let \hat{p} be the maximum p such that (x_1^p, y_1^p) is undeleted, i.e. $\hat{p} = \max\{p \mid i_p = 1\}$. Analogously, let \check{p} be the minimum p such that (x_4^p, y_4^p) is undeleted, i.e. $\check{p} = \min\{p \mid i_p = 4\}$. Note that both exist, since we doubled the arcs (x_1^{-1}, y_1^{-1}) and (x_4^{a+1}, y_4^{a+1}) . So the cycle $C_{\hat{p}, \check{p}}$ exists and has non-negative weight $0 \leq 2A - (A + (\hat{p} + 2)) - (A - (\check{p} - 1)) = \check{p} - \hat{p} - 3$. This is equivalent to $\hat{p} + 3 \leq \check{p}$. So for every index $\hat{p} < p < \check{p}$, both arcs (x_1^p, y_1^p) and (x_4^p, y_4^p) are deleted and there are at least two such indices.

Next we show that $(x_2^{\hat{p}+1}, y_2^{\hat{p}+1})$ is also deleted. Otherwise, we have that $i_{\hat{p}+1} = 2$ and thus the cycle $C_{\hat{p}, \hat{p}+1}$ has weight $2A - (A + (\hat{p} + 2)) - (A - (\hat{p} + 1)) = -1$, which is a contradiction to $C_{\hat{p}, \hat{p}+1}$ having non-negative weight. So we have $i_{\hat{p}+1} = 3$. Now define \tilde{p} to be the maximum p such that (x_3^p, y_3^p) is undeleted, i.e. $\tilde{p} = \max\{p \mid i_p = 3\}$. By the previous argument we know that this maximum exists and $\hat{p} + 1 \leq \tilde{p}$.

Consider the cycle $C_{\tilde{p}, \hat{p}}$. Since this cycle again has non-negative weight, we have $0 \leq 2A - (A + (\tilde{p} - 1)) - (A - (\hat{p} + 1)) = \tilde{p} - \hat{p} - 2$. This is equivalent to $\tilde{p} + 2 \leq \hat{p}$. Note that for the index $\tilde{p} + 1$, we have that $\hat{p}, \tilde{p} < \tilde{p} + 1 < \check{p}$. That means the only arc that can be undeleted for $\tilde{p} + 1$ is $(x_2^{\tilde{p}+1}, y_2^{\tilde{p}+1})$, i.e. $i_{\tilde{p}+1} = 2$. We want to show that $p^* = \tilde{p} + 1$ is indeed the only index p with $i_p = 2$. For this let $\overleftarrow{p}^* = \min\{p \mid i_p = 2\}$ and $\overrightarrow{p}^* = \max\{p \mid i_p = 2\}$. By the weight of $C_{\overleftarrow{p}^*, \tilde{p}}$ being non-negative, we get $0 \leq 2A - (A - \overleftarrow{p}^*) - (A + (\tilde{p} + 1)) = \overleftarrow{p}^* - \tilde{p} - 1$, i.e. $\tilde{p} + 1 \leq \overleftarrow{p}^*$.

Now consider any $i \in \{2, 3\}$ and any $p \in \{-1, 0, \dots, a+1\}$. By Claim 4.7, we have that S contains either $(\dot{x}_i^p, \dot{y}_i^p)$ or $(\ddot{x}_i^p, \ddot{y}_i^p)$, but none of the arcs (y_i^p, \dot{x}_i^p) and (\dot{y}_i^p, x_i^p) . Thus, if (x_i^p, y_i^p) is undeleted, by the negative cycle $(x_i^p, y_i^p) \circ (y_i^p, \dot{x}_i^p) \circ (\dot{x}_i^p, \dot{y}_i^p) \circ (\dot{y}_i^p, x_i^p)$, we have that $(\dot{x}_i^p, \dot{y}_i^p)$ is deleted and thus $(\ddot{x}_i^p, \ddot{y}_i^p)$ is undeleted. So for \tilde{p} , the arc $(\ddot{x}_3^{\tilde{p}}, \ddot{y}_3^{\tilde{p}})$ is undeleted, and for \overrightarrow{p}^* the arc $(\ddot{x}_2^{\overrightarrow{p}^*}, \ddot{y}_2^{\overrightarrow{p}^*})$ is undeleted. By Claim 4.7, we also get

that none of the other arcs of the following cycle $\vec{C}_{\vec{p}^*, \tilde{p}}$ are deleted:

$$(\tilde{s}_2, \tilde{x}_2^{\vec{p}^*}) \circ (\tilde{x}_2^{\vec{p}^*}, \tilde{y}_2^{\vec{p}^*}) \circ (\tilde{y}_2^{\vec{p}^*}, \tilde{t}_2) \circ (\tilde{t}_2, \tilde{s}_3) \circ (\tilde{s}_3, \tilde{x}_3^{\tilde{p}}) \circ (\tilde{x}_3^{\tilde{p}}, \tilde{y}_3^{\tilde{p}}) \circ (\tilde{y}_3^{\tilde{p}}, \tilde{t}_3) \circ (\tilde{t}_3, \tilde{s}_2) .$$

As this cycle exists in $R^{a,b} - S$, it has non-negative weight, which implies

$$0 \leq -(A + \vec{p}^*) + A - (A - (\tilde{p} + 1)) + A = \tilde{p} - \vec{p}^* + 1 .$$

This is equivalent to $\vec{p}^* \leq \tilde{p} + 1$. Overall we have $\tilde{p} + 1 \leq \overleftarrow{p}^* \leq \vec{p}^* \leq \tilde{p} + 1$. So indeed the only index p with (x_2^p, y_2^p) undeleted is $p^* = \tilde{p} + 1$.

Finally, consider the $v_{\text{in}}^r \rightarrow v_{\text{out}}^r$ -paths in $R^{a,b} - S$ for $r \in \{1, \dots, b\}$. Note that each such path starts with the subpath $v_{\text{in}}^r, g_{\text{in}}^r, x_2^p, y_2^p$ and ends with the (potentially overlapping) subpath $x_2^{p'}, y_2^{p'}, g_{\text{out}}^r, v_{\text{out}}^r$ for $p, p' \in \{-1, 0, \dots, a+1\}$. As the only index with (x_2^p, y_2^p) undeleted is $p^* = \tilde{p} + 1$, we have that the only $v_{\text{in}}^r \rightarrow v_{\text{out}}^r$ -path in $R^{a,b} - S$ has the form $v_{\text{in}}^r, g_{\text{in}}^r, x_2^{p^*}, y_2^{p^*}, g_{\text{out}}^r, v_{\text{out}}^r$. This path has weight

$$-(A - p^*) - (A + p^*) - M + \varphi_r(p^*) + 2A + M = \varphi_r(p^*) .$$

It only remains to show that $p^* \in \{1, \dots, a\}$. For this note that $-1 \leq \hat{p} \leq \tilde{p} - 1$ which is equivalent to $1 \leq \tilde{p} + 1 = p^*$, and that $p^* = \tilde{p} + 1 \leq \tilde{p} - 1 \leq a$. \blacksquare

After proving what minimum solutions look like for our meta gadget, we can now give the overall reduction from CLIQUE. Let v_1, \dots, v_n be an arbitrary ordering of $V(G)$ and e_1, \dots, e_m be an arbitrary ordering of $E(G)$. For every $i \in \{1, \dots, k\}$, we introduce a vertex gadget H^i which is a copy of the meta gadget $R^{a,b}$ with $a = n$, $b = 2$, $M = n$ and the functions $\varphi_1(p) = p$ and $\varphi_2(p) = -p$. Moreover, we rename the vertices v_{in}^1 and v_{out}^1 of H_i to s and z_i^+ , and the vertices v_{in}^2 and v_{out}^2 of H_i to s and z_i^- . For every $i, j \in \{1, \dots, k\}$ with $i < j$, we introduce an edge gadget $H^{i,j}$ which is a copy of the meta gadget $R^{a,b}$ with $a = m$, $b = 4$, $M = n$ and the functions defined as follows. For every $e_\ell = \{v_p, v_q\} \in E(G)$ with $p \leq q$ we let $\varphi_1(\ell) = -p$, $\varphi_2(\ell) = p$, $\varphi_3(\ell) = -q$ and $\varphi_4(\ell) = q$. Moreover, we rename the vertices in the following way:

- v_{in}^1 and v_{out}^1 of $H_{i,j}$ become z_i^+ and t ,
- v_{in}^2 and v_{out}^2 of $H_{i,j}$ become z_i^- and t ,
- v_{in}^3 and v_{out}^3 of $H_{i,j}$ become z_j^+ and t ,
- v_{in}^4 and v_{out}^4 of $H_{i,j}$ become z_j^+ and t ,

Any renamed vertices sharing the same name are identified with each other. Finally, we add an arc (t, s) of weight 0. Call the resulting graph H . We claim that H as NEGATIVE DFAS instance has a solution of size at most $d = k \cdot 5(n + 3) + \frac{1}{2}k(k + 1) \cdot 5(m + 3)$ if and only if G has a clique of size k .

For the forward direction let S be any solution to H as NEGATIVE DFAS instance of size at most d . By Claim 4.7, we know that S restricted to any H^i has size at least $5(n + 3)$ and restricted to any $H^{i,j}$ has size at least $5(m + 3)$. By d matching exactly

the total of these numbers, we know that S has exactly these sizes in the gadgets. Moreover, (t, s) is not contained in S .

Using Claim 4.9, we get that for every H^i there is a $p_i \in \{1, \dots, n\}$ such that any $s \rightarrow z_i^+$ -path has weight p_i and any $s \rightarrow z_i^-$ -path has weight $-p_i$. Also, we get that for every $H^{i,j}$ there is an $\ell_{i,j}$ such that for $e_{\ell_{i,j}} = \{v_p, v_q\}$ we have that any $z_i^+ \rightarrow t$ -path has weight $-p$, any $z_i^- \rightarrow t$ -path has weight p , any $z_j^+ \rightarrow t$ -path has weight $-q$ and any $z_j^- \rightarrow t$ -path has weight q . Let now

$$V' = \{v_{p_i} \mid i \in \{1, \dots, k\}\} \subseteq V(G) \text{ and } E' = \{e_{\ell_{i,j}} \mid i, j \in \{1, \dots, k\}, i < j\} \subseteq E(G) .$$

We claim that (V', E') is a clique in G of size k . As V' and E' have exactly the right cardinality, we only have to proof that for any unordered pair $v_{p_i}, v_{p_j} \in V'$ there is an edge between them. Without loss on generality we can assume $i < j$, otherwise swap the indices. Moreover, let $e_{\ell_{i,j}} = \{v_p, v_q\}$.

Consider the cycle consisting of the arc (t, s) , the $s \rightarrow z_i^+$ -path in $H^i - S$ and the $z_i^+ \rightarrow t$ -path in $H^{i,j} - S$. This cycle is non-negative and has weight $p_i - p$, thus $p \leq p_i$. By considering the cycle consisting of the arc (t, s) , the $s \rightarrow z_i^-$ -path in $H^i - S$ and the $z_i^- \rightarrow t$ -path in $H^{i,j} - S$, we get a non-negative cycle of weight $-p_i + p$. Thus, $p \geq p_i$ and combined with the former inequality, $p = p_i$. Analogously, by using the paths starting and ending in z_j^+ and z_j^- instead, we get that $p_j = q$. Thus, we get that indeed the edge $\{v_{p_i}, v_{p_j}\} = \{v_p, v_q\} = e_{\ell_{i,j}} \in E(G)$. So G contains the clique (V', E') of size k .

For the backward direction consider any clique on vertices v_{p_1}, \dots, v_{p_k} in G . Choose by Claim 4.8 a solution S^i of size $5(n+3)$ for every gadget H^i such that any $s \rightarrow z_i^+$ -path has weight p_i and any $s \rightarrow z_i^-$ -path has weight $-p_i$. Analogously, by Claim 4.8 choose a solution $S^{i,j}$ of size $5(n+3)$ for every gadget $H^{i,j}$ such that any $z_i^+ \rightarrow t$ -path has weight $-p_i$, any $z_i^- \rightarrow t$ -path has weight p_i , any $z_j^+ \rightarrow t$ -path has weight $-p_j$ and any $z_j^- \rightarrow t$ -path has weight p_j . Note that the latter solution can be chosen that way, because $\{v_{p_i}, v_{p_j}\} \in E(G)$ and $i < j$.

We claim that $S = \bigcup_i S^i \cup \bigcup_{i < j} S^{i,j}$ is a solution to H of size d . The size bound follows from adding up the sizes of the individual solutions. It remains to check that $H - S$ contains no negative cycles. As S constraint to a single gadget was chosen as a solution, the only negative cycles that can exists use some high-level vertices. There are two different types of such cycles. The first type uses the arc (t, s) , followed by an $s \rightarrow z_i^+$ -path in H^i for some i and an $z_i^+ \rightarrow t$ -path in some $H^{i,j}$ or $H^{j,i}$. Our partial solutions were chosen such that these cycles have weight $0 + p_i - p_i$. The other type uses the arc (t, s) , followed by an $s \rightarrow z_i^-$ -path in H^i for some i and an $z_i^- \rightarrow t$ -path in some $H^{i,j}$ or $H^{j,i}$. But by choice of S these paths have weight $0 - p_i + p_i$. So $H - S$ contains no negative cycles.

It remains to show that the treedepth and the number of arcs with positive weight is bounded. First we will derive bounds for these parameters on the meta-gadgets $R^{a,b}$. For the treedepth we will use the recursive formula of Definition 2.8. Consider the set

$$D = \{s^1, s^2, s^3, s^4, t^1, t^2, t^3, t^4, \check{s}_2, \check{s}_3, \check{t}_2, \check{t}_3\} \cup \{g_{\text{in}}^r, g_{\text{out}}^r, v_{\text{in}}^r, v_{\text{out}}^r \mid r \in \{1, \dots, b\}\} .$$

Note that $R^{a,b} - D$ consists only of the mini-gadgets, which are connected components of size 16 each. Thus, $\text{td}(R^{a,b}) \leq |D| + 16 = 28 + 4b$.

For the number of arcs with positive weight note that A and M are chosen such that the only positive arc weights are A and $2A + M$. Of the former there are ten and of the later there are b many in every meta gadget. Thus, the number of arcs with positive weight in $R^{a,b}$ is bounded by $10 + b$.

For the graph H , we can combine the above bounds to bound its treedepth and its number of positive arcs. Regarding the treedepth of H , note that for $D' = \{s, t\} \cup \{z_i^+, z_i^- \mid i \in \{1, \dots, k\}\}$, the graph $H - D'$ consists of connected components that only form subgraphs of the H^i and $H^{i,j}$. By Definition 2.8, it holds

$$\begin{aligned} \text{td}(H) &\leq |D'| + \text{td}(H - D') \leq |D'| + \sum_{i=1}^k \left(\text{td}(H^i) + \sum_{j=i+1}^k \text{td}(H^{i,j}) \right) \\ &\leq |D'| + 36k + 22k(k+1) \leq 22k^2 + 60k + 2. \end{aligned}$$

Regarding the number of positive arcs in H , note that we did not add any new arc of positive weight. Thus, the number of positive arcs in H is bounded by $12k + 7k(k+1)$. \square

4.4 W[1]-hardness for Pathwidth, Deletion Size and Few Positive Arcs

In this section we prove W[1]-hardness for MINFB with general integral weights when parameterized in $\text{pw}(G) + k + w_+$. To this end, we make a reduction from MULTICOLORED CLIQUE to NEGATIVE DFAS. In addition, we will see how this implies W[1]-hardness for MINFB parameterized in $\text{pw}(G) + k$ for instances with weights of the form $w : A(G) \rightarrow \{-1, 0, 1\}$.

Theorem 4.10. *NEGATIVE DFAS is W[1]-hard when parameterized by the pathwidth, deletion size and number of positive arcs. This holds already for instances (G, w, k) , where $w : A(G) \rightarrow \mathbb{Z} \cap [-|V(G)|^2, |V(G)|^2]$.*

Proof. We prove W[1]-hardness by showing a parameterized reduction from MULTICOLORED CLIQUE. For this purpose, let (G, k) an instance of MULTICOLORED CLIQUE with $V(G) = \dot{\bigcup}_{i=1}^k V^i$. For ease of notation, define sets $E^{i,j}$ as the set of edges of type $\{v_i, v_j\}$ with $v_i \in V^i$ and $v_j \in V^j$, for $i, j = 1, \dots, k$ with $i < j$. Moreover, we abuse this notation slightly by defining $E^{j,i} = E^{i,j}$ but with the order of vertices switched, i.e. $\{v_i, v_j\} \in E^{i,j}$ but $\{v_j, v_i\} \in E^{j,i}$.

We construct an equivalent instance (H, w, d) of NEGATIVE DFAS, see Figure 5 for an illustration. The graph H will have three types of gadgets:

1. a vertex gadget H^i representing V^i for every $i \in \{1, \dots, k\}$,
2. an edge gadget $H^{i,j}$ representing $E^{i,j}$ for every $1 \leq i < j \leq k$, and
3. a consistency gadget C^i for every $i \in \{1, \dots, k\}$.

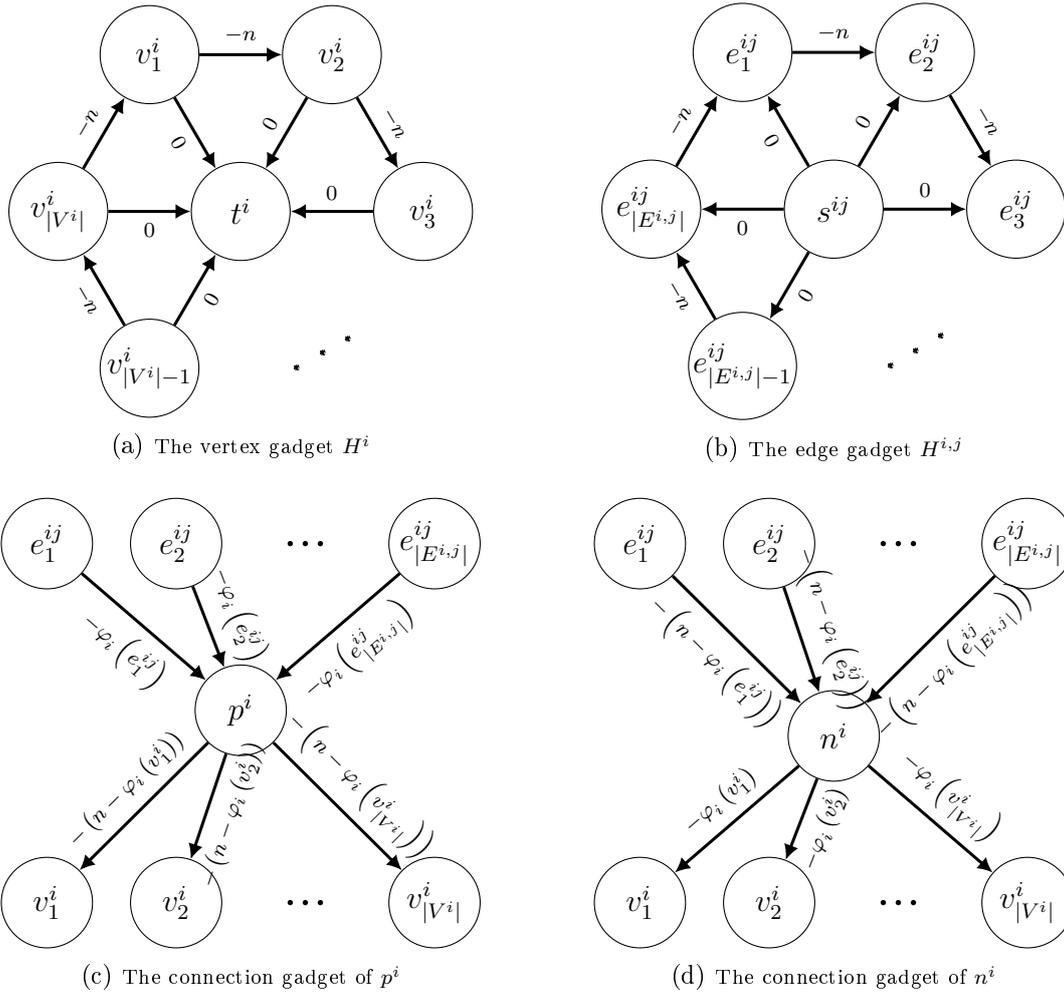


Figure 5: Gadgets used in the construction of Theorem 4.10

The gadgets H^i representing $V^i = \{v_1^i, \dots, v_{|V^i|}^i\}$ consist of one vertex t^i and the vertices of V^i . Each gadget H^i contains two types of arcs. First, it has arcs (v_r^i, t^i) of weight 0 for every $v_r^i \in V^i$, and second, arcs (v_r^i, v_{r+1}^i) of weight $-n$ for every $r \in \{1, \dots, |V^i|\}$, where we define $v_{|V^i|+1}^i = v_1^i$.

The gadgets $H^{i,j}$ representing $E^{i,j} = \{e_1^{i,j}, \dots, e_{|E^{i,j}|}^{i,j}\}$ consist of one vertex $s^{i,j}$ and one vertex for every $e_r^{i,j} \in E^{i,j}$. Each gadget $H^{i,j}$ contains two types of arcs. First, it has arcs $(s^{i,j}, e_r^{i,j})$ of weight 0 for every $e_r^{i,j} \in E^{i,j}$, and second, it has arcs $(e_r^{i,j}, e_{r+1}^{i,j})$ of weight $-n$ for every $r \in \{1, \dots, |E^{i,j}|\}$, where we define $e_{|E^{i,j}|+1}^{i,j} = e_1^{i,j}$.

Finally, the consistency gadgets C_i consist of the vertices n^i and p^i and interconnect the existing gadgets as follows. First there is an arc $(t^i, s^{i,j})$ of weight $n(|V^i| + |E^{i,j}| - 1)$ for every $j \in \{1, \dots, k\} \setminus \{i\}$. Then we need a bijective function $\varphi_i : V^i \rightarrow \{1, \dots, |V^i|\}$ (for example, $\varphi_i(v_r^i) = r$ will do). We abuse the notation and that every edge of E has at most one endpoint in every V^i and define for $e^{i,j} = \{v^i, v^j\} \in E^{i,j}$ that $\varphi_i(e^{i,j}) = \varphi_i(v^i)$. With this definition in mind, we add the following arcs to C^i :

- $(e^{i,j}, n^i)$ of weight $-(n - \varphi_i(e^{i,j}))$ for every $j \in \{1, \dots, k\} \setminus \{i\}$ and $e^{i,j} \in E^{i,j}$,
- $(e^{i,j}, p^i)$ of weight $-\varphi_i(e^{i,j})$ for every $j \in \{1, \dots, k\} \setminus \{i\}$ and $e^{i,j} \in E^{i,j}$,
- (n^i, v^i) of weight $-\varphi_i(v^i)$ for every $v^i \in V^i$, and
- (p^i, v^i) of weight $-(n - \varphi_i(v^i))$ for every $v^i \in V^i$.

Let H denote the resulting graph consisting of all the gadgets, and let w denote the corresponding weight function. By choosing $d = k + \binom{k}{2}$, we obtain our NEGATIVE DFAS instance (H, w, d) .

Negative DFAS to Multicolored Clique. Let S be a solution to (H, w, d) . The gadgets H^i and $H^{i,j}$ each contain a negative cycle consisting of the arcs (v_r^i, v_{r+1}^i) or $(e_r^{i,j}, e_{r+1}^{i,j})$, respectively. As there are exactly $k + \binom{k}{2}$ such gadgets and their cycles are disjoint, S has size exactly d and contains exactly one arc $(v_{r_i}^i, v_{r_i+1}^i)$ for every $i \in \{1, \dots, k\}$ and one arc $(e_{r_{i,j}}^{i,j}, e_{r_{i,j}+1}^{i,j})$ for every $1 \leq i < j \leq k$. Define $K = (\{v_r^i \mid i \in \{1, \dots, k\}, r \in \{2, \dots, |V^i| + 1\}, (v_{r-1}^i, v_r^i) \in S\}, \{e_r^{i,j} \mid 1 \leq i < j \leq k, r \in \{1, \dots, |E^{i,j}|\}, (e_r^{i,j}, e_{r+1}^{i,j}) \in S\})$. We claim that K is a solution to (G, k) i.e. a multicolored clique in G . As K contains exactly one element of every V^i and $E^{i,j}$, we only have to check that the edges really connect to the vertices of K .

Suppose, for sake of contradiction, that this is not the case, i.e. there is a vertex $v^i \in V^i \cap V(K)$ and an edge $e^{i,j} \in E^{i,j} \cap E(K)$ for some $1 \leq i, j \leq k, i \neq j$, such that $e^{i,j} = \{u^i, w^j\}$ with $u^i \neq v^i$. For $e^{i,j}$ we have deleted only the outgoing arc in gadget $H^{i,j}$, so there is an $s^{i,j} \rightarrow e^{i,j}$ -path P of length $|E^{i,j}|$ that uses one arc of weight 0 and $(|E^{i,j}| - 1)$ many arcs of weight $-n$. So, $w(P) = -n(|E^{i,j}| - 1)$.

Likewise, we deleted only the incoming arc of v^i in the gadget H^i . Therefore, there is an $v^i \rightarrow t^i$ -path Q of length $|V^i|$ that uses one arc of weight 0 and $(|V^i| - 1)$ many arcs of weight $-n$. Thus, $w(Q) = -n(|V^i| - 1)$. We link P and Q together with the arc $(t^i, s^{i,j})$ of weight $n(|V^i| + |E^{i,j}| - 1)$ to an overall $v_i \rightarrow e^{i,j}$ -path $R = Q \circ (t^i, s^{i,j}) \circ P$ of weight $n(|V^i| + |E^{i,j}| - 1) - n(|E^{i,j}| - 1) - n(|V^i| - 1) = n$.

Let us now consider the unique $e^{i,j} \rightarrow v^i$ -paths

$$W^+ = \{(e^{i,j}, p^i), (p^i, v^i)\} \text{ and } W^- = \{(e^{i,j}, n^i), (n^i, v^i)\}$$

in the consistency gadget C_i . These are using only the internal vertices p^i or n^i , respectively. Their weights are

$$\begin{aligned} w(W^+) &= -(n - \varphi_i(v^i)) - \varphi_i(e^{ij}) = (\varphi_i(v^i) - \varphi_i(u^i)) - n & \text{and} \\ w(W^-) &= -(n - \varphi_i(e^{ij})) - \varphi_i(v^i) = (\varphi_i(u^i) - \varphi_i(v^i)) - n . \end{aligned}$$

If we join these two paths with R , we obtain two cycles with weights

$$\begin{aligned} w(W^+ \circ R) &= (\varphi_i(v^i) - \varphi_i(u^i)) - n + n = \varphi_i(v^i) - \varphi_i(u^i) & \text{and} \\ w(W^- \circ R) &= (\varphi_i(u^i) - \varphi_i(v^i)) - n + n = \varphi_i(u^i) - \varphi_i(v^i) . \end{aligned}$$

As $v^i \neq u^i$ and φ_i is bijective, we have that either $w(W^+ \circ R)$ or $w(W^- \circ R)$ is negative. Thus, $R \circ W^+$ or $R \circ W^-$ is a negative cycle, a contradiction to $H - S$ containing no negative cycles. Hence, K must be consistent and a clique in G .

Multicolored Clique to Negative DFAS. Now suppose that K is a multicolored clique on k vertices in G , i.e. a solution to (G, k) . We choose S to be the set $\{(v_{r_{i-1}}^i, v_{r_i}^i) \mid i \in \{1, \dots, k\}, v_{r_i}^i \in V^i \cap V(K)\} \cup \{(e_{r_{i,j}}^{i,j}, e_{r_{i,j}+1}^{i,j}) \mid 1 \leq i < j \leq k, e_{r_{i,j}}^{i,j} \in E^{i,j} \cap E(K)\}$. Note that now every gadget H^i and $H^{i,j}$ is acyclic on its own. Furthermore, the whole graph H without our set S and the arcs $(t^i, s^{i,j})$ is acyclic.

Claim 4.11. *Every $s^{i,j} \rightarrow t^{i'}$ -path in $H - S$ has weight at least $-n(|V^{i'}| + |E^{i,j}| - 1)$.*

Proof of Claim 4.11. Suppose, for sake of contradiction, that there is an $s^{i,j} \rightarrow t^{i'}$ -path in $H - S$ of weight lower than $-n(|V^{i'}| + |E^{i,j}| - 1)$. Let P the shortest such path with respect to the number of arcs. Then P does not use any arc of type $(t^p, s^{p,q})$, as they have weight $n(|V^p| + |E^{p,q}| - 1)$ and therefore either $w(P[s^{i,j}, t^p]) < -n(|V^p| + |E^{i,j}| - 1)$ or $w(P[s^{p,q}, t^{i'}]) < -n(|V^{i'}| + |E^{p,q}| - 1)$ and those have fewer arcs. But as P does not use any arc of type $(t^p, s^{p,q})$ it can only go from the gadget $H^{i,j}$ to $H^{i'}$ once with the help of $C^{i'}$ and does not use other gadgets. Moreover, we have that $i' \in \{i, j\}$. We can assume, without loss of generality, that $i' = i$. Then P leaves $H^{i,j}$ exactly once and enters H^i exactly once. Let $e_a^{i,j}$ be the last vertex of $H^{i,j}$ and v_b^i the first vertex of H^i our path P visits. Note that $w(P[s^{i,j}, e_a^{i,j}])$ and $w(P[v_b^i, t^i])$ are always multiples of n and have weight at least $-n(|E^{i,j}| - 1)$ and $-n(|V^i| - 1)$, respectively. Also, this weight is only attained if and only if $a = r_{i,j}$ or $b = r_i$, respectively.

In any case, $P[e_a^{i,j}, v_b^i]$ consists of three vertices with n^i or p^i being the middle one. By choice of weights in C^i , the lowest weight $P[e_a^{i,j}, v_b^i]$ can achieve is $1 - |V^i| - n > -2n$. Also, if $a = r_{i,j}$ and $b = r_i$ the weight is $-n$. So in any case the composition $P = P[s^{i,j}, e_a^{i,j}] \circ P[e_a^{i,j}, v_b^i] \circ P[v_b^i, t^i]$ has weight at least $-n(|V^i| + |E^{i,j}| - 1)$ —a contradiction to the choice of P . This completes the proof of the claim. \blacksquare

As every cycle in $H - S$ has to use an arc $(t^i, s^{i,j})$ of weight $n(|V^i| + |E^{i,j}| - 1)$ and every path completing this arc to a cycle has weight at least $-n(|V^{i'}| + |E^{i,j}| - 1)$, the graph $H - S$ contains no negative cycle.

For the size bound note that S contains exactly one arc from every H^i and $H^{i,j}$. Thus, it has size $k + \binom{k}{2} = d$. So S is a solution to (H, w, d) .

Bounding the pathwidth of H . The pathwidth of H is defined as the pathwidth of the underlying undirected graph \bar{H} . We will show that $\text{pw}(\bar{H}) \in \mathcal{O}(k^2)$.

Note that, if for some set $U \subseteq V(\bar{H})$ we get a path decomposition of $\bar{H} - U$ with bag size at most b , we get a path decomposition for \bar{H} with bag sizes at most $b + |U|$ by adding U to every bag. Thus, $\text{pw}(\bar{H}) \leq \text{pw}(\bar{H} - U) + |U|$. Let

$$U = \bigcup_{i=1}^k \{t^i, n^i, p^i, v_{|V^i|}^i\} \cup \bigcup_{1 \leq i < j \leq k} \{s^{i,j}, e_{|E^{i,j}|}^{i,j}\}.$$

This set U has size $4k + k(k - 1) = k^2 + 3k$. Observe that the graph $\bar{H} - U$ consist exactly of paths of the following form. Each vertex gadget H^i turned into a path consisting of the edges $\{v_\ell^i, v_{\ell+1}^i\}$ for $\ell \in \{1, \dots, |V^i| - 2\}$. Meanwhile, each edge gadget $H^{i,j}$ turned into a path consisting of the edges $\{e_\ell^{i,j}, v_{\ell+1}^{i,j}\}$ for $\ell \in \{1, \dots, |E^{i,j}| - 2\}$.

Furthermore, note that if we have a path decomposition for every connected component, then we can concatenate them to one path decomposition. Thus, the pathwidth of $\bar{H} - U$ is the maximum pathwidth of its connected components. But these are all paths and have pathwidth one. Thus, $\text{pw}(\bar{H}) \leq \text{pw}(\bar{H} - U) + |U| = k^2 + 3k + 1 \in \mathcal{O}(k^2)$.

Overall we have proven that our reduction produces an instance whose pathwidth and deletion size are bounded by some function in the parameter of the original MULTICOLORED CLIQUE instance. Moreover, by choice of φ , the only arcs of positive weight are the arcs of type $(t^i, s^{i,j})$ of which only $k(k - 1)$ many exist. So we have a parameterized reduction from MULTICOLORED CLIQUE with parameter k to NEGATIVE DFAS with parameters pathwidth, deletion size and number of positive arcs.

To see that the weights have the claimed form, first note that all of them are integral. Moreover, the highest absolute weight have the arcs $(t^i, s^{i,j})$, which have weight $n(|V^i| + |E^{i,j}| - 1)$. As $|V(H)| \geq |V^i| + |E^{i,j}|$ and $|V(H)| \geq n$, we have that indeed $w : A(H) \rightarrow \mathbb{Z} \cap [-|V(H)|^2, |V(H)|^2]$, proving the theorem. \square

From this theorem we are also able to infer W[1]-hardness for NEGATIVE DFAS parameterized in $\text{pw}(G) + k$ on instances with weights restricted to $w : A(G) \rightarrow \{-1, 0, 1\}$.

Theorem 4.12. *NEGATIVE DFAS is W[1]-hard when parameterized in the pathwidth and deletion size, even on instances with weights of the form $w : A(G) \rightarrow \{-1, 0, 1\}$.*

Proof. We proof the theorem by showing how an instance described by Theorem 4.10 can be turned into one with its pathwidth increased by two and weights of the form $w : A(G) \rightarrow \{-1, 0, 1\}$ while keeping the deletion size the same. In the process, it loses the property of having only a bounded number of positive arcs. Let (G, w, k) be a NEGATIVE DFAS instance as described in Theorem 4.10. Especially, we have

$w : A(G) \rightarrow \mathbb{Z} \cap [-|V(G)|^2, |V(G)|^2]$. From this instance we obtain an instance (G', w', k) by replacing every non-zero arc $a \in A(G)$ by a path P_a of $|w(a)|$ -many arcs all having weight 1 if $w(a) > 0$ or -1 if $w(a) < 0$. We denote by (G', w') the resulting graph with arc weights. It remains to show that (G, w, k) and (G', w', k') are solution equivalent and $\text{pw}(G') \leq \text{pw}(G) + 2$.

To see the solution equivalence, note that any cycle in G' uses a newly created path P_a either as a whole or not at all. Thus, deleting an arbitrary arc of a path P_a is equivalent to deleting the whole path and this is equivalent to deleting the arc a in G . For the pathwidth, note that replacing arcs by paths can be incorporated into the path-decomposition as follows. Let B_i the bag that contains the arc a in the path decomposition of G that defines $\text{pw}(G)$. Let $x_1, \dots, x_{|w(a)|-1}$ the internal vertices of P_a . Then we modify the path decomposition of G by replacing B_i with the sequence $B_i, B_i \cup \{x_1, x_2\}, \dots, B_i \cup \{x_{|w(a)|-2}, x_{|w(a)|-1}\}, B_i$. This covers all newly created arcs for P_a . By doing this replacement for all arcs a (while using only B_i of the original decomposition), this yields a new path decomposition for G' , where the maximum bag size increased by at most two. This shows that $\text{pw}(G') \leq \text{pw}(G) + 2$, concluding the proof of the theorem. \square

4.5 W[1]-hardness for Pathwidth, Deletion Size and Few Negative Arcs

In this section we are going to prove W[1]-hardness for MINFB parameterized in w_-, k and $\text{pw}(G)$ for general arc weights. We will do this by doing an intermediate step showing W[1]-hardness for the BOUNDED EDGE DIRECTED (s, t) -CUT problem in DAGs when parameterized in k and $\text{pw}(G)$.

BOUNDED EDGE DIRECTED (s, t) -CUT

Input: A graph G , vertices $s, t \in V(G)$ and two integers $k, \ell \in \mathbb{Z}_{\geq 0}$.

Task: Find a set $S \subseteq V(G)$ of size at most k such that every $s \rightarrow t$ -path of $G - S$ has length more than ℓ or decide that no such set exists.

For the undirected BOUNDED EDGE (s, t) -CUT problem, Bentert et al. showed W[1]-hardness for the parameters maximum vertex degree and pathwidth [3, Theorem 1]. By inspecting their reduction closely, one easily sees that their proof actually shows hardness for parameters maximum vertex degree, pathwidth and deletion size. The hardness for the directed case follows from this by replacing every edge by a forward and backward arc. For completeness (and bound on the pathwidth) we choose to do a direct reduction to the directed version here. It has the nice property that it produces acyclic digraphs, a property we cannot achieve by replacing arcs with a *cycle* of length two. However, note that all important ideas are already present in the paper by Bentert et al. [3].

Lemma 4.13. *The BOUNDED EDGE DIRECTED (s, t) -CUT problem parameterized in the pathwidth $\text{pw}(G)$, deletion size d and maximum degree of the underlying undirected graph $\Delta(G)$ is W[1]-hard, even when the graph is restricted to be acyclic.*

Proof. Let now (G, k) be a CLIQUE instance. Recall that CLIQUE is W[1]-hard when parameterized by k . We want to construct an equivalent instance (H, d, Λ) of BOUNDED EDGE DIRECTED (s, t) -CUT whose parameters $\text{pw}(H)$, d and $\Delta(G)$ are bounded by a function of k only. For notation, we fix some arbitrary ordering v_1, \dots, v_n of the vertices $V(G)$. For the edges $e \in E(G)$ we define their canonical representation $\{v_i, v_j\}$ to be the order of vertices with $i \leq j$. Setting this canonical representation then defines an ordering $e_1 \prec \dots \prec e_m$ of the edges $E(G)$ by $e = \{v_i, v_j\} \preceq e' = \{v_{i'}, v_{j'}\}$ iff $(i, j) \preceq_{\text{lex}} (i', j')$, where \preceq_{lex} is the lexicographical ordering on vectors of length two.

As usual in reduction from cliques we construct two kinds of gadgets, vertex gadgets H^i and edge gadgets $H^{i,j}$. The vertex gadgets each choose a vertex in the graph and the edge gadgets check whether there is indeed an edge between them. For all these constructions we will use paths of different lengths. For ease of notation we will replace these paths by arcs with positive integral weights. We make sure that these weights are bounded by some function of $n + m$ and that parallel arcs have weight at least two. In this way, replacing an arc of weight w by a directed path of length w gives us a still polynomial sized simple graph for BOUNDED EDGE DIRECTED (s, t) -CUT. Also, deletion of an arc of a path whose internal vertices have in-degree and out-degree one and are disjoint from s and t , means that no arc on this path is part of an $s \rightarrow t$ -path anymore. So a solution does not delete more paths by choosing several arcs on the same replaced path. This preserves solution equivalence, between the weighted and the replaced graph.

An important role for the weights plays an integer M which we will choose later. Our choice of Λ will depend on M , so we also defer its choice until later. For now think of M as a big integer.

Gadgets. Before introducing the different vertex and edge gadgets, we discuss some gadget that will be the foundation for both. For any positive integers $a, b \in \mathbb{Z}_{>0}$ and any set $\{\varphi^q : \{1, \dots, b\} \rightarrow \mathbb{Z} \cap [-M, M] \mid q \in \{1, \dots, a\}\}$ of functions, we introduce the gadget $R^{a,b}$. The gadget $R^{a,b}$ consists of a many disjoint paths P^1, \dots, P^a of length $2b + 2$. For each path P^q with $q \in \{1, \dots, a\}$ all arcs of P^q have weight 1, and we denote the vertices of P^q in order of their appearance by $p_{\text{in}}^q, p_1^q, \hat{p}_1^q, p_2^q, \hat{p}_2^q, \dots, p_{b+1}^q, p_{\text{out}}^q$. We add the following “detours” along the paths P^q . For every $r \in \{1, \dots, b\}$ there is an arc (p_r^q, p_{r+1}^q) of weight $5M + \varphi^q(r)$. Moreover, we interlink the paths P^q and P^{q+1} for every $q \in \{1, \dots, a-1\}$ by the following arcs. For every $r \in \{1, \dots, b\}$ there are arcs $(\hat{p}_r^q, p_{r+1}^{q+1})$ and $(\hat{p}_r^{q+1}, p_{r+1}^q)$ of weight $3M$ each. Finally, the gadget $R^{a,b}$ contains the vertices s and t . To connect those vertices to the rest of the graph, for every $q \in \{1, \dots, a\}$, there are two parallel arcs (s, p_1^q) of weight 2 and two parallel arcs (p_{b+1}^q, t) of weight $\Lambda - 4M$. This concludes the construction of our gadget. Note that for $M \geq 1$ the graph has positive arc weights and for $\Lambda \geq 4M + 2$ all parallel arcs are of weight at least 2.

We now prove the crucial properties of this gadget:

Claim 4.14. *Let $M \geq 2b + 2$ and $\Lambda \geq 4M + 2$ be integers. For every set $S \subseteq A(R^{a,b})$ such that $R^{a,b} - S$ contains no $s \rightarrow t$ -path of weight $\leq \Lambda$, we have that $|S| \geq a$.*

Moreover, for every such set $S \subseteq A(R^{a,b})$ of size $|S| = a$, there exists a unique $x \in \{1, \dots, b\}$ such that for every $q \in \{1, \dots, a\}$ any minimum-weight $s \rightarrow p_{\text{out}}^q$ -path has weight $5M + 2b + 1 + \varphi^q(x)$ and any minimum-weight $p_{\text{in}}^q \rightarrow t$ -path has weight $\Lambda + M + 2b - 1 + \varphi^q(x)$.

Also, for every $x \in \{1, \dots, b\}$, we can choose a set $S \subseteq A(R^{a,b})$ of size a such that $R^{a,b} - S$ contains no $s \rightarrow t$ -path of weight $\leq \Lambda$, any minimum-weight $s \rightarrow p_{\text{out}}^q$ -path has weight $5M + 2b + 1 + \varphi^q(x)$ and any minimum-weight $p_{\text{in}}^q \rightarrow t$ -path has weight $\Lambda + M + 2b - 1 + \varphi^q(x)$.

Proof of Claim 4.14. Consider for every $q \in \{1, \dots, a\}$ the path $s, p_1^q, \hat{p}_1^q, \dots, p_{b+1}^q, t$. This path has weight $2 + 2b + \Lambda - 4M \leq \Lambda$. Thus, any set S as in the theorem statement has to contain an arc of all of these $s \rightarrow t$ -paths. As these paths are arc-disjoint, we have that all sets have size at least a .

Now we turn to the sets S of size exactly a . Consider the paths from above again. As there are two parallel arcs for each of (s, p_1^q) and (p_{b+1}^q, t) , our set S has to intersect each P^q with exactly one edge that is not (p_{in}^q, p_1^q) or $(p_{b+1}^q, p_{\text{out}}^q)$.

We claim that there is an $x \in \{1, \dots, b\}$ such that $S = \{(p_x^q, \hat{p}_x^q) \mid q \in \{1, \dots, a\}\}$. Suppose that not, then for some pair (p, q) with $q \in \{1, \dots, a - 1\}$ and $y \in \{1, \dots, n\}$ we have that either $P^q[p_1^q, \hat{p}_y^q]$ and $P^{q+1}[p_{y+1}^{q+1}, p_{b+1}^{q+1}]$ are disjoint from S or $P^{q+1}[p_1^{q+1}, \hat{p}_y^{q+1}]$ and $P^q[p_{y+1}^q, p_{b+1}^q]$ are disjoint from S . In the former case, the path

$$(s, p_1^q) \circ P^q[p_1^q, \hat{p}_y^q] \circ (\hat{p}_y^q, p_{y+1}^{q+1}) \circ P^{q+1}[p_{y+1}^{q+1}, p_{b+1}^{q+1}] \circ (p_{b+1}^{q+1}, t)$$

is an $s \rightarrow t$ -path in $R^{a,b} - S$. In the latter case, the path

$$(s, p_1^{q+1}) \circ P^{q+1}[p_1^{q+1}, \hat{p}_y^{q+1}] \circ (\hat{p}_y^{q+1}, p_{y+1}^q) \circ P^q[p_{y+1}^q, p_{b+1}^q] \circ (p_{b+1}^q, t)$$

is an $s \rightarrow t$ -path in $R^{a,b} - S$. So one of these $s \rightarrow t$ -paths exists in $R^{a,b} - S$. Note that both paths have weight

$$2 + (2(y - 1) + 1) + 3M + 2(b - y) + (\Lambda - 4M) = 2b + 1 + 3M + \Lambda - 4M \leq \Lambda,$$

which is a contradiction to the choice of S .

So, if there is a set S such that $R^{a,b} - S$ contains no $s \rightarrow t$ -path of weight $\leq \Lambda$ that has size a , it has the form $\{(p_x^q, \hat{p}_x^q) \mid q \in \{1, \dots, a\}\}$. Indeed, for such a set S , we have that any $s \rightarrow t$ -path in $R^{a,b} - S$ has to use one of the detour arcs (p_x^q, p_{x+1}^q) of weight $5M + \varphi^q(x) \geq 4M$. Moreover, any $s \rightarrow t$ -path has to use an arc incident to s and t of weight 2 and $\Lambda - 4M$ respectively. So for any $x \in \{1, \dots, b\}$, each $s \rightarrow t$ -path in $R^{a,b} - \{(p_x^q, \hat{p}_x^q) \mid q \in \{1, \dots, a\}\}$ has overall weight at least $2 + 4M + \Lambda - 4M > \Lambda$.

It only remains to prove that, for solutions $S = \{(p_x^q, \hat{p}_x^q) \mid q \in \{1, \dots, a\}\}$, any minimum-weight $s \rightarrow p_{\text{out}}^q$ -path has weight $5M + 2b + 1 + \varphi^q(x)$ and any minimum-weight $p_{\text{in}}^q \rightarrow t$ -path has weight $\Lambda + M + 2b - 1 + \varphi^q(x)$ in $R^{a,b} - S$. Any $s \rightarrow p_{\text{out}}^q$ -path uses exactly one incident edge of s (which has weight 2) and p_{out}^q (which has weight 1). So, for these paths it is enough to prove that any $p_1^h \rightarrow p_{b+1}^q$ -path in $R^{a,b} - S$

for $h \in \{1, \dots, a\}$ has weight at least $5M + 2(b - 1) + \varphi^q(x)$ and one path achieves that weight. Moreover, any $p_{\text{in}}^q \rightarrow t$ -path uses exactly one incident edge of p_{in}^q (which has weight 1) and t (which has weight $\Lambda - 4M$). Thus, for these paths it is enough to prove that any $p_1^q \rightarrow p_{b+1}^g$ -path in $R^{a,b} - S$ for $g \in \{1, \dots, a\}$ has weight at least $5M + 2(b - 1) + \varphi^q(x)$ and one path achieves that weight. Together it suffices to show that any $p_1^h \rightarrow p_{b+1}^g$ -path in $R^{a,b} - S$ has weight at least $5M + 2(b - 1) + \varphi^q(x)$ and there is a $p_1^q \rightarrow p_{b+1}^g$ -path that achieves this weight.

Notice that any $p_1^h \rightarrow p_{b+1}^g$ -path contains a detour arc (p_x^r, p_{x+1}^r) of weight $5M + \varphi^h(x) \geq 4M$. If a $p_1^h \rightarrow p_{b+1}^g$ -path contains another detour arc or an interconnecting arc, it occurs an additional weight of at least $3M$. So any $p_1^h \rightarrow p_{b+1}^g$ -path that has uses arcs other than those of a single P^q and the arc (p_x^q, p_{x+1}^q) has weight at least $8M$. But $5M + 2b + 1 + \varphi^q(x) \leq 7M < 8M$. So it is enough to show that there is a $p_1^q \rightarrow p_{b+1}^g$ -path of weight $5M + 2(b - 1) + \varphi^q(x)$. Notice that $P^q[p_1^q, p_x^q] \circ (p_x^q, p_{x+1}^q) \circ P^q[p_{x+1}^q, p_{b+1}^g]$ has weight $2(x - 1) + 5M + \varphi^q(x) + 2(b - x) = 5M + 2(b - 1) + \varphi^q(x)$ and thus is exactly such a path. This completes the proof of Claim 4.14. \blacksquare

We are now ready to define the vertex gadgets and edge gadgets. All vertex gadgets will be identical and all edge gadgets will be identical, just the interconnection between them differs. For all $i \in \{1, \dots, k\}$, we introduce a vertex gadget H^i by setting $a = 2$, $b = n$, $\varphi^1(x) = x$ and $\varphi^2(x) = -x$. Note that the image of φ^1 and φ^2 lies in $\mathbb{Z} \cap [-M, M]$ for $M \geq n$. Thus, the graph $R^{2,n}$ is well-defined. To obtain our gadget H^i from this we forget about the vertices p_{in}^1 and p_{in}^2 . Furthermore, we rename the vertices p_{out}^1 and p_{out}^2 to v_+^i and v_-^i .

For all $i, j \in \{1, \dots, k\}$ with $i < j$, we introduce an edge gadget $H^{i,j}$. For this we set $a = 4$, $b = m$ and for every $x \in \{1, \dots, m\}$ and edge e_x with canonical representation (v_g, v_h) , we set $\varphi^1(x) = -g$, $\varphi^2(x) = g$, $\varphi^3(x) = -h$ and $\varphi^4(x) = h$. Again the image of $\varphi^1, \dots, \varphi^4$ lies in $\mathbb{Z} \cap [-M, M]$ for $M \geq m$. Thus, the graph $R^{4,m}$ is well-defined. By forgetting the vertices p_{out}^q and renaming $p_{\text{in}}^1, p_{\text{in}}^2, p_{\text{in}}^3$ and p_{in}^4 to v_+^i, v_-^i, v_+^j and v_-^j .

From these gadgets we obtain our complete graph H by taking one copy of each gadget and identifying all vertices of the same name (s, t, v_+^i and v_-^i). Finally, we set $d = 2n + 2n(n - 1)$, $M = 2(n + m) + 2$ and $\Lambda = 10M + 2(n - 1) + 2(m - 1) - 1$. This completes our construction.

Correctness. We now want to prove that (G, k) as instance of CLIQUE is a “yes”-instance if and only if (H, d, Λ) as instance of BOUNDED EDGE DIRECTED (s, t) -CUT is a “yes”-instance.

For the forward direction let $U \subseteq V(G)$ be a clique in G with corresponding edge set F . Let u_1, \dots, u_k be the indices of the vertices of v_1, \dots, v_n corresponding to the vertices of U in increasing order. Also, let $f_{i,j}$ be the index of the edge $\{v_{u_i}, v_{u_j}\}$ in the fixed ordering of $E(G)$. By Claim 4.14, there is a solution S^i of size two to each of the H^i 's such that each $s \rightarrow v_+^i$ -path has weight at least $5M + 2(n - 1) + u_i$ and each $s \rightarrow v_-^i$ -path has weight at least $5M + 2(n - 1) - u_i$. Moreover, again by Claim 4.14, there is a solution $S^{i,j}$ of size four to each of the $H^{i,j}$'s such that each $v_+^i \rightarrow t$ -path has weight at least $5M + 2(m - 1) - u_i$, each $v_-^i \rightarrow t$ -path has weight at least $5M + 2(m - 1) + u_i$, each $v_+^j \rightarrow t$ -path has weight at least $5M + 2(m - 1) - u_j$ and each $v_-^j \rightarrow t$ -path has weight at least $5M + 2(m - 1) + u_j$. We define our solution S as

$\bigcup_{i=1}^k S^i \cup \bigcup_{1 \leq i < j \leq k} S^{i,j}$. By definition our S is a solution when restricted to a single vertex gadget or a single edge gadget. The only $s \rightarrow t$ -paths that are not contained in a single gadget consist of an $s \rightarrow v_+^i$ -path in some vertex gadget and a $v_+^i \rightarrow t$ -path in some edge gadget or of an $s \rightarrow v_-^i$ -path in some vertex gadget and a $v_-^i \rightarrow t$ -path in some edge gadget. In each case, as seen above, these paths have length at least $5M + 2(n - 1) + u_i + 5M + 2(m - 1) - u_i = \Lambda + 1$. It only remains to check the size of S which is exactly $2k + 4 \frac{k(k-1)}{2} = d$.

For the backwards direction, let S be a solution to (H, d, Λ) . From Claim 4.14, we get that any solution to a vertex gadget has size at least two and any solution to an edge gadget has size at least four. As a solution must be a solution to every single gadget and our gadgets are arc disjoint, we have that any solution must have size at least $2k + 4 \frac{k(k-1)}{2} = d$. From this it follows that our solution must have size exactly two in every vertex gadget and size exactly four in every edge gadget. Claim 4.14 states that the structure of these solutions in the vertex gadgets must be such that in $H^i - S$ any minimum weight $s \rightarrow v_+^i$ -path in $V^i - S$ has weight $5M + 2(n - 1) + u_i$ and any minimum weight $s \rightarrow v_-^i$ -path in V^i has weight $5M + 2(n - 1) - u_i$ for some $u_i \in \{1, \dots, n\}$. We define the vertex set $U \subseteq V(G)$ to be the set $\{v_{u_i} \mid i \in \{1, \dots, k\}\}$.

We can apply Claim 4.14 also to the structure of S with respect to $H^{i,j} - S$. This tells us that for every $1 \leq i < j \leq k$ there is an index $f_{i,j} \in \{1, \dots, m\}$ such that for the canonical representation $\{v_g, v_h\}$ of $e_{f_{i,j}}$, in $H^{i,j} - S$ we have that any minimum weight $v_+^i \rightarrow t$ -path has weight $5M + 2(m - 1) - g$, any minimum weight $v_-^i \rightarrow t$ -path has weight $5M + 2(m - 1) + g$, any minimum weight $v_+^j \rightarrow t$ -path has weight $5M + 2(m - 1) - h$ and any minimum weight $v_-^j \rightarrow t$ -path has weight $5M + 2(m - 1) + h$. If we concatenate a minimum weight $s \rightarrow v_+^i$ -path in $V^i - S$ with a minimum weight $v_+^i \rightarrow t$ -path in $H^{i,j} - S$, we get an $s \rightarrow t$ -path of weight $10M + 2(n - 1) + 2(m - 1) + u_i - g$. As S is a solution to (H, d, Λ) , this path has length more than Λ , which is equivalent to $u_i + 1 > g$ or $u_i \geq g$. By applying the same argument to minimum weight paths that concatenate at v_-^i instead of v_+^i , we get the inequality $u_i \leq g$. Together this yields $u_i = g$. If we swap H^i for H^j and v_+^i and v_-^i for v_+^j and v_-^j , we get $u_j = h$ in the same way. So the edge $\{v_{u_i}, v_{u_j}\} = \{v_g, v_h\} = e_{f_{i,j}}$ exists in G . Since this is true for all $1 \leq i < j \leq k$, U is indeed the vertex set of a clique.

Bound on Parameters and Acyclicity. For pathwidth consider the vertex set $U = \{s, t\} \cup \bigcup_{i=1}^k \{v_+^i, v_-^i\}$. This is a set of $2(k + 1)$ vertices and will be contained in every bag of our path decomposition. Note that all vertices of $H - U$ have no neighbors outside the gadget they are contained in. So it suffices to find a path decomposition for every gadget (minus vertices in U), concatenate them and add U to every set. The pathwidth of this decomposition is then bounded by $|U|$ plus the maximum pathwidth of any gadget path decomposition. We do this path decomposition on the meta-gadget $R^{a,b}$. For every $r \in \{1, \dots, b\}$ let $X_{2r-1} = \{p_r^q, \hat{p}_r^q \mid q \in \{1, \dots, a\}\}$ and for every $r \in \{1, \dots, b - 1\}$ let $X_{2r} = \{\hat{p}_r^q, p_{r+1}^q \mid q \in \{1, \dots, a\}\}$. Then every arc of $R^{a,b} - S$ is contained in one of the bags X_i and each vertex in at most two consecutive bags. Thus, X_1, \dots, X_{2b-1} is a path decomposition. The maximum size of a bag is $2a$. For our specific gadgets a is at most four. Thus, we get that the pathwidth of H

is at most $8 + |U| = 2k + 10$. The maximum vertex degree inside a gadget is 6 for the p_r^a vertices. However, the vertices v_+^i , v_-^i , s and t have neighbors in several gadgets. Nodes v_+^i , v_-^i have degree $k + \frac{k(k-1)}{2}$, s has degree $4k$ and t has degree $16\frac{k(k-1)}{2}$. In any case the maximum is bounded by a function of k . Last but not least the deletion size is $d = 2k + 4\frac{k(k-1)}{2}$.

For the acyclicity note that the gadgets $R^{a,b}$ are acyclic (as all arcs only run to increasing positions along the paths P^a). So it only remains to prove that the boundary vertices s , t , v_+^i and v_-^i are connected in a way that preserves acyclicity. Node s has only outgoing arcs and t has only incoming arcs. Moreover, all v_+^i and v_-^i the ingoing arcs come exactly from the vertex gadgets and the outgoing arcs come exactly from the edge gadgets and these gadgets are not connected by any other vertices. \square

Now we use Lemma 4.13 to show hardness for NEGATIVE DFAS by weighting all arcs of an BOUNDED EDGE DIRECTED (s, t) -CUT instance with weight one and introducing $(k + 1)$ -many (t, s) arcs of weight $-(\ell + 1)$.

Theorem 4.15. *NEGATIVE DFAS is W[1]-hard when parameterized in the pathwidth, deletion size and number of negative arcs.*

Proof. We show the theorem by reduction from BOUNDED EDGE DIRECTED (s, t) -CUT on DAGs parameterized by pathwidth and deletion size, which is W[1]-hard by Lemma 4.13. Let (G, s, t, k, ℓ) be a BOUNDED EDGE DIRECTED (s, t) -CUT instance with G being a DAG. We construct a solution equivalent instance (H, w, k) of NEGATIVE DFAS as follows. To construct H , give all arcs in G weight 1 and introduce $k + 1$ arcs (t, s) of weight $-(\ell + 1)$. This completes the construction of (H, w, k) . We now show the solution equivalence.

Bounded Edge Directed (s, t) -Cut to Negative Directed Feedback Arc Set.

Let S be a solution to (G, s, t, k, ℓ) . We claim that S is also a solution to (H, w, k) . The set S trivially fulfills the size bound of k . Suppose, for sake of contradiction, that there is a negative cycle C in $H - S$. As the only negative arcs of H are those from t to s , C contains exactly one of these. So $C[s, t]$ is an $s \rightarrow t$ -path in $G - S$ and has weight less than $-w((t, s)) = \ell + 1$. Now, all weights in G are 1 and thus $C[s, t]$ is an $s \rightarrow t$ -path in $G - S$ of length at most ℓ , a contradiction to S being a solution to (G, s, t, k, ℓ) . Thus, S is a solution to (H, w, k) .

Negative DFAS to Bounded Edge Directed (s, t) -Cut. Let S be an inclusion-wise minimal solution to (H, w, k) . As there are $(k + 1)$ -many parallel arcs (t, s) , one of them does not lie in S . Any cycle in H can use at most one of these arcs and thus as S is inclusion-wise minimal, S contains non of them, i.e. it contains only arcs of G . We claim that S is a solution to (G, s, t, k, ℓ) . It trivially fulfills the size bound. Now assume for contradiction that $G - S$ contains an $s \rightarrow t$ -path P of length at most ℓ . Then $P \circ (t, s)$ is a cycle of weight $w(P) - (\ell + 1) < 0$ in $H - S$, which is a contradiction to S being a solution to (H, w, k) .

Bounding the parameters. The deletion size of our new instance stays the same and thus is bounded by the deletion size of the old instance. The same holds for the number of negative arcs, as we only introduced $k + 1$ of them. Finally, note that we

can turn any pathwidth decomposition of G into one of H by adding $\{s, t\}$ to all bags. This shows $\text{pw}(H) \leq \text{pw}(G) + 2$. \square

4.6 No Admission of a Polynomial Compression

In this section, we exhibit that there is no polynomial compression for MINFB parameterized by $k + w_-$. We give a reduction from the BOUNDED EDGE DIRECTED (s, t) -CUT problem. Recall by last section that this problem takes as input a digraph G , vertices s, t , and integers $k, \ell \in \mathbb{N}$, and asks for a set $X \subseteq E(G)$ of size at most k such that $G - X$ contains no s - t -paths of length at most ℓ . Fluschnik et al. [24] showed that BOUNDED EDGE DIRECTED (s, t) -CUT does not admit a kernel of size polynomial in k and ℓ , assuming $\text{NP} \not\subseteq \text{coNP}/\text{poly}$, even for acyclic input digraphs. In fact, their construction allows for a stronger result, ruling out a polynomial compression.

Proposition 4.16 (Fluschnik et al. [24]). *Assuming $\text{NP} \not\subseteq \text{coNP}/\text{poly}$, BOUNDED EDGE DIRECTED (s, t) -CUT does not admit a polynomial compression in $k + \ell$ even when G is a DAG.*

To get to the MINFB problem, we first consider as an intermediate step the DIRECTED SMALL CYCLE TRANSVERSAL problem.

DIRECTED SMALL CYCLE TRANSVERSAL

Input: A directed graph G and integers k, ℓ .

Task: Find a set $X \subseteq E(G)$ of size at most k such that $G - X$ contains no cycles of length at most ℓ .

Fluschnik et al. [24] showed that DIRECTED SMALL CYCLE TRANSVERSAL does not admit a kernel of size polynomial in k and ℓ , unless $\text{NP} \subseteq \text{coNP}/\text{poly}$. Again their result can be strengthened to not admitting a polynomial compression. They further observed that DIRECTED SMALL CYCLE TRANSVERSAL admits a simple branching algorithm that runs in time $\mathcal{O}(\ell^k \cdot n \cdot (n + m))$. Here we argue that a dependence on both parameters k and ℓ is necessary for fixed-parameter tractability:

Lemma 4.17. *There is a polynomial parameter transformation from BOUNDED EDGE DIRECTED (s, t) -CUT in DAGs with parameter k (parameter ℓ , parameter $k + \ell$) to DIRECTED SMALL CYCLE TRANSVERSAL with parameter k (resp. ℓ , resp. $k + \ell$).*

This even holds if DIRECTED SMALL CYCLE TRANSVERSAL is restricted to instances where there are vertices $s, t \in V(G)$ such that all cycles of G consist of an s - t -path and an arc of the form (t, s) .

Proof. Let (G, s, t, k, ℓ) be a BOUNDED EDGE DIRECTED (s, t) -CUT instance where G is a DAG. As G is a DAG, it admits a topological ordering $v_1, \dots, v_{|V(G)|}$ of its vertices, so that there are no arcs (v_i, v_j) for $j < i$. Without loss of generality, let

$v_1 = s$ and $v_{|V(G)|} = t$, as vertices before s or after t in a topological ordering are never part of any s - t -path.

We now create a directed graph G' from G by adding $k+1$ parallel arcs a_1, \dots, a_{k+1} from t to s . Then every cycle in G' consists of an s - t -path and an arc a_i , as G was acyclic. Set $\ell' = \ell + 1$. Then (G', k, ℓ') is an instance of DIRECTED SMALL CYCLE TRANSVERSAL. The above transformation can be done in polynomial time and the parameter increases by at most one (depending on whether ℓ is part of the parameter).

It remains to show that (G', k, ℓ') is a “yes”-instance of DIRECTED SMALL CYCLE TRANSVERSAL if and only if (G, s, t, k, ℓ) is a “yes”-instance of BOUNDED EDGE DIRECTED (s, t) -CUT.

For the forward direction, let X' be a solution to (G', k, ℓ') . Consider $X = X' \setminus \{a_1, \dots, a_{k+1}\}$. For sake of contradiction, suppose that $G - X$ contains an s - t -path P of length at most ℓ . As $|X'| \leq k$, it can not contain all arcs a_i . Without loss of generality, $a_1 \notin X'$. Then P followed by a_1 is a cycle in $G' - X'$ of length at most $\ell + 1 = \ell'$ —a contradiction to X' being a solution to (G', k, ℓ') .

For the reverse direction, let X be a solution to (G, s, t, k, ℓ) . Then X is also a solution to (G', k, ℓ') by the following argument. Suppose, for sake of contradiction, that $G' - X$ contains a cycle C of length at most ℓ' . By the structure of G' , C consists of an s - t -path P in $G - X$ and an arc a_i . Then $|P| = |C| - 1 \leq \ell$, contradicting that X is a solution to (G, k, ℓ) . \square

Lemma 4.18. *There is a PPT from DIRECTED SMALL CYCLE TRANSVERSAL on instances where every cycle uses an arc of type (t, s) when parameterized by k (by $k + \ell$) to NEGATIVE DFAS parameterized by k (resp. $k + w_-$, where w_- is the number of arcs with negative weight); this even holds in the case where $w : A(G) \rightarrow \{\pm 1\}$.*

Proof. We start with a DIRECTED SMALL CYCLE TRANSVERSAL instance (G, k, ℓ) as described in the lemma. Let a_1, \dots, a_p be the arcs of the form (t, s) . For any $p \geq k + 1$ there is always an arc which survives the deletion of some arc set of at most k elements. So we can assume $p \leq k + 1$ as deleting superfluous arcs does not change the solution. Set $A_{+1} = A(G) \setminus \{a_1, \dots, a_p\}$. Now replace the a_i by mutually disjoint (except for s and t) paths P_i of length ℓ . Call the resulting directed graph G' and let $A_{-1} = \cup_{i=1}^p A(P_i)$. Finally, define $w(a) = 1$ for $a \in A_{+1}$ and $w(a) = -1$ for $a \in A_{-1}$. As $A(G') = A_{-1} \uplus A_{+1}$, the function $w : A(G') \rightarrow \{-1, +1\}$ is well defined.

The instance (G', w, k) has the required form. Also the transformation can be made in polynomial time. As k remains unchanged and $w_- = |A_{-1}| = \ell \cdot p \leq \ell \cdot (k + 1)$ is bounded by a polynomial in $k + \ell$, the parameter restrictions of PPTs are fulfilled. It remains to prove that (G', w, k) is a “yes”-instance of NEGATIVE DFAS if and only if (G, k, ℓ) is a “yes”-instance of DIRECTED SMALL CYCLE TRANSVERSAL.

For the forward direction, let X' be a solution of (G', w, k) . Let X be the set where every arc of X' which is part of some P_i is replaced by a_i (and duplicates are removed). Clearly, $|X| \leq |X'| \leq k$. Suppose there is a cycle C of length at most ℓ in $G - X$. Then C contains a unique arc a_j . Let C' be the cycle in G' resulting from the replacement of a_j by P_j in C . Then C' is also in $G' - X'$ by choice of X and contains at most $\ell - 1$ arcs in A_{+1} and ℓ arcs in A_{-1} . This yields the contradiction $w(C') = |A(C') \cap A_{+1}| - |A(C') \cap A_{-1}| \leq \ell - 1 - \ell = -1$.

For the reverse direction, let X be a solution of (G, k, ℓ) . Let X' the set X where every arc a_i is replaced by the first arc of P_i . By definition of G' , $X' \subseteq A(G')$ and $|X'| = |X| \leq k$ holds. Now suppose there is a cycle C' in $G' - X'$ with $w(C') < 0$. As the paths P_i are mutually disjoint (except the end vertices) every inner vertex of each P_i has in-degree and out-degree one. Thus, if there is an arc of some P_i inside C' the whole path P_i is. Replace each such P_i by a_i to obtain a cycle C . By construction of G' and X' , this cycle C is in $G - X$. Each cycle in G and therefore also C contains exactly one arc of type a_i . Therefore, C' contains exactly one path P_i and from $0 > w(C') = |A(C') \cap A_{+1}| - |A(C') \cap A_{-1}| = |A(C') \cap A_{+1}| - \ell$ we get that $|A(C') \cap A_{+1}| < \ell$. As $|A(C') \cap A_{+1}|$ is integral we can sharpen the bound to $|A(C') \cap A_{+1}| \leq \ell - 1$. By $A(C) = (A(C') \cap A_{+1}) \uplus \{a_i\}$, we get that $|A(C)| = |A(C') \cap A_{+1}| + 1 \leq \ell$ —a contradiction. \square

Concatenating all reductions above and combing with Theorem 2.3, we obtain the following corollary:

Corollary 4.19. *There is a PPT from BOUNDED EDGE DIRECTED (s, t) -CUT parameterized by k (in $k + \ell$) to MINFB parameterized by k (in $k + w_-$). This even holds for instances of MINFB where A is a system of difference constraints and $w \in \{\pm 1\}^m$.*

This corollary yields the desired result.

Theorem 4.20. *Assuming $\text{NP} \not\subseteq \text{coNP}/\text{poly}$, MINFB does not admit a polynomial compression when parameterized by $k + w_-$ even for systems A of difference constraints and right-hand sides $w \in \{\pm 1\}^m$.*

Proof. This follows by combining Proposition 4.16 and Corollary 4.19 with the help of Proposition 2.11. \square

5 Discussion

We investigated the MINFB problem from the perspective of parameterized complexity. We settled the complexity of the problem for various choices of parameters, such as the size of a deletion set or the numbers of positive and negative right-hand sides. In addition, structural parameters were contemplated: treewidth, pathwidth and treedepth. We also ruled out the existence of a polynomial compression for combined parameters $k + w_-$, assuming that $\text{coNP} \not\subseteq \text{NP}/\text{poly}$.

It remains a challenging open problem whether DFAS admits a polynomial compression for parameter k , and whether our perspective from the more general MINFB problem can help in answering that. A polynomial compression for MINFB for parameter k or $k + w_+$, even for node-arc incidence matrices, would imply a polynomial compression for DFAS (as the reduction from DFAS to MINFB does not increase the parameter).

Another interesting open question is whether the problem is tractable when parameterized by pathwidth whenever the pathwidth is at most 5. Recall that in Section 4.2, we show that the problem is NP-hard when the parameter pathwidth has value 6.

Acknowledgement. Kristóf Bérczi and Lydia Mirabel Mendoza-Cadena were supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number FK128673. “Application Domain Specific Highly Reliable IT Solutions” project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Sub-programme) funding scheme. Alexander Göke was partially supported by DFG grant MN 59/1-1. The research was supported by the DAAD with funds from the Bundesministerium für Bildung und Forschung.

References

- [1] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoret. Comput. Sci.*, 209(1-2):237–260, 1998.
- [2] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. System Sci.*, 54(2, part 2):317–331, 1997.
- [3] M. Bentert, K. Heeger, and D. Knop. Length-bounded cuts: Proper interval graphs and structural parameters. *J. Comput. Syst. Sci.*, 126:21–43, 2022.
- [4] P. Berman and M. Karpinski. Approximating minimum unsatisfiability of linear equations. In *Proc. SODA 2002*, pages 514—516, 2002.
- [5] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.
- [6] H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.
- [7] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995.
- [8] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.
- [9] H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theoret. Comput. Sci.*, 412(35):4570–4578, 2011.
- [10] M. Bonamy, Ł. Kowalik, J. Nederlof, M. Pilipczuk, A. Socała, and M. Wrochna. On directed feedback vertex set parameterized by treewidth. In A. Brandstädt, E. Köhler, and K. Meer, editors, *Proc. WG 2018*, pages 65–78, 2018.

-
- [11] N. Chakravarti. Some results concerning post-infeasibility analysis. *European J. Oper. Res.*, 73(1):139–143, 1994.
- [12] T. F. N. Chan, J. W. Cooper, M. Koutecký, D. Král’, and K. Pekárková. Matrices of optimal tree-depth and row-invariant parameterized algorithm for integer programming. In *Proc. ICALP 2020*, volume 168 of *Leibniz Int. Proc. Informatics*, pages 26:1–26:19, 2020.
- [13] J. Chen, Y. Liu, S. Lu, B. O’Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5):Art. 21, 19, 2008.
- [14] J. W. Chinneck. An effective polynomial-time heuristic for the minimum-cardinality IIS set-covering problem. *Ann. Math. Artif. Intell.*, 17(1):127–144, 1996.
- [15] J. W. Chinneck. Finding a useful subset of constraints for analysis in an infeasible linear program. *INFORMS J. Comput.*, 9(2):164–174, 1997.
- [16] J. W. Chinneck. *Feasibility and infeasibility in optimization: algorithms and computational methods*, volume 118 of *International Series in Operations Research & Management Science*. Springer, New York, USA, 2008.
- [17] R. Chitnis, M. Cygan, M. Hajiaghayi, and D. Marx. Directed subset feedback vertex set is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(4):Art. 28, 28, 2015.
- [18] G. Codato and M. Fischetti. Combinatorial benders’ cuts for mixed-integer linear programming. *Oper. Res.*, 54(4):756–766, 2006.
- [19] R. Crowston, G. Gutin, M. Jones, and A. Yeo. Parameterized complexity of satisfying almost all linear equations over \mathbb{F}_2 . *Theory Comput. Syst.*, 52(4):719–728, 2013.
- [20] M. Cygan, F. Fomin, B. M. Jansen, Ł. Kowalik, D. Lokshtanov, Dániel Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Open problems collected for the 2014 School on Parameterized Complexity in Będlewo, Poland, 2014. <http://fptschool.mimuw.edu.pl/opl.pdf>.
- [21] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer, Cham, 2015.
- [22] M. Cygan, Ł. Kowalik, and M. Pilipczuk. Open problems from 2013 Workshop on Kernels, 2013. <http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf>.
- [23] R. W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

-
- [24] T. Fluschnik, D. Hermelin, A. Nichterlein, and R. Niedermeier. Fractals for kernelization lower bounds, with an application to length-bounded cut problems. In *Proc. ICALP 2016*, volume 55 of *Leibniz Int. Proc. in Informatics*, pages 25:1–25:14, 2016.
- [25] F. V. Fomin, D. Lokshantov, S. Saurabh, M. Pilipczuk, and M. Wrochna. Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. *ACM Trans. Algorithms*, 14(3):34:1–34:45, 2018.
- [26] T. Gallai. Maximum-minimum Sätze über Graphen. *Acta Mathematica Hungarica*, 9(3-4):395–434, 1958.
- [27] R. Ganian, T. Peitl, F. Slivovsky, and S. Szeider. Fixed-parameter tractability of dependency QBF with structural parameters. In *Proc. KR 2020*, pages 392–402, 2020.
- [28] P. Giannopoulos, C. Knauer, and G. Rote. The parameterized complexity of some geometric problems in unbounded dimension. In *Proc. IWPEC 2009*, volume 5917 of *Lecture Notes Comput. Sci.*, pages 198–209. 2009.
- [29] A. Göke. *Vertex Deletion Problems: A Parameterized Point of View*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany, November 2020.
- [30] A. Göke, L. M. Mendoza-Cadena, and M. Mnich. Resolving infeasibility of linear systems: A parameterized approach. In *Proc. IPEC 2019*, volume 148 of *Leibniz Int. Proc. Informatics*, pages 17:1–17:15, 2019.
- [31] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, Berlin, second edition, 1993.
- [32] S. Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optim.*, 8(1):61–71, 2011.
- [33] Y. Iwata, T. Ogasawara, and N. Ohsaka. On the power of tree-depth for fully polynomial FPT algorithms. In *Proc. STACS 2018*, volume 96 of *Leibniz Int. Proc. Informatics*, pages 41:1–41:14, 2018.
- [34] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM, Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, USA, 1972.
- [35] T. Kloks. *Treewidth: computations and approximations*, volume 842. Springer Science & Business Media, USA, 1994.
- [36] S. Kratsch and M. Wahlström. Two edge modification problems without polynomial kernels. *Discrete Optim.*, 10(3):193–199, 2013.

-
- [37] N. Leithäuser, S. O. Krumke, and M. Merkert. Approximating infeasible 2vpi-systems. In *Proc. WG 2012*, volume 7551 of *Lecture Notes Comput. Sci.*, pages 225–236, 2012.
- [38] D. Lokshтанov, M. S. Ramanujan, and S. Saurabh. When recursion is better than iteration: A linear-time algorithm for acyclicity with few error vertices. In *Proc. SODA 2018*, pages 1916–1933, 2018.
- [39] M. Mnich and E. J. van Leeuwen. Polynomial kernels for deletion to classes of acyclic digraphs. *Discrete Optim.*, 25:48–76, 2017.
- [40] J. Nešetřil and P. O. De Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European J. Combinatorics*, 27(6):1022–1041, 2006.
- [41] M. E. Pfetsch. Branch-and-cut for the maximum feasible subsystem problem. *SIAM J. Optim.*, 19(1):21–38, 2008.
- [42] G. Ramalingam, J. Song, L. Joskowicz, and R. E. Miller. Solving systems of difference constraints incrementally. *Algorithmica*, 23(3):261–275, 1999.
- [43] B. Roy. Transitivité et connexité. *Comptes Rendus Hebdomadaires Des Seances De L Academie Des Sciences*, 249(2):216–218, 1959.
- [44] J. K. Sankaran. A note on resolving infeasibility in linear programs by constraint relaxation. *Oper. Res. Lett.*, 13(1):19–20, 1993.
- [45] K. Subramani and P. Wojciechowski. A combinatorial certifying algorithm for linear feasibility in UTVPI constraints. *Algorithmica*, 78(1):166–208, 2017.
- [46] S. Warshall. A theorem on Boolean matrices. *J. ACM*, 9(1):11–12, 1962.
- [47] Open problems from 2010 Workshop on Kernels, 2010. <http://fpt.wdfiles.com/local--files/open-problems/open-problems.pdf>.
- [48] M. Zehavi. Mixing color coding-related techniques. In *Proc. ESA 2015*, volume 9294 of *Lecture Notes Comput. Sci.*, pages 1037–1049, 2015.