On Making a Distinguished Vertex Minimum Degree by Vertex Deletion

Nadja Betzler[1,⋆], Robert Bredereck[2,⋆], Rolf Niedermeier[2], Johannes Uhlmann[1,⋆⋆]

[1]Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
[2]Institut für Softwaretechnik und Theoretische Informatik,
Technische Universität Berlin, Franklinstr. 28/29, D-10587 Berlin, Germany
{nadja.betzler, robert.bredereck, johannes.uhlmann}@uni-jena.de
rolf.niedermeier@tu-berlin.de

Abstract. For directed and undirected graphs, we study the problem to make a distinguished vertex the unique minimum-(in)degree vertex through deletion of a minimum number of vertices. The corresponding NP-hard optimization problems are motivated by applications concerning control in elections and social network analysis. Continuing previous work for the directed case, we show that the problem is W[2]-hard when parameterized by the graph's feedback arc set number, whereas it becomes fixed-parameter tractable when combining the parameters "feedback vertex set number" and "number of vertices to delete". For the so far unstudied undirected case, we show that the problem is NP-hard and W[1]-hard when parameterized by the "number of vertices to delete". On the positive side, we show fixed-parameter tractability for several parameterizations measuring tree-likeness, including a vertex-linear problem kernel with respect to the parameter "feedback edge set number". On the contrary, we show a non-existence result concerning polynomial-size problem kernels for the combined parameter "vertex cover number and number of vertices to delete", implying corresponding nonexistence results when replacing vertex cover number by treewidth or feedback vertex set number.

1 Introduction

Making a distinguished vertex minimum degree by vertex deletion is a natural though widely unexplored graph problem. We contribute new insights into the algorithmic complexity of the corresponding computational problems, providing intractability as well as fixed-parameter tractability results.

Formally, we studied the following two decision problems.

⋆ Supported by the DFG, research project PAWS, NI 369/10.
⋆⋆ Supported by the DFG, research project PABI, NI 369/7.

Published in the Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, Nový Smokovec, Slovakia, January 2011, Volume 6543 in Lecture Notes in Computer Science, pp. 123–134.* © *Springer*

MIN-INDEGREE DELETION (MID)
Given: A directed graph $D = (W, A)$, a distinguished vertex $w_c \in W$, and an integer $k \geq 1$.
Question: Is there a subset $W' \subseteq W \setminus \{w_c\}$ of size at most k such that w_c is the only vertex that has minimum indegree in $D[W \setminus W']$?

Whereas MID has been studied in one previous paper [2], its undirected counterpart seems completely unexplored:

MIN-DEGREE DELETION (MDD)
Given: An undirected graph $G = (V, E)$, a distinguished vertex $w_c \in V$, and an integer $k \geq 1$.
Question: Is there a subset $V' \subseteq V \setminus \{w_c\}$ of size at most k such that w_c is the only vertex that has minimum degree in $G[V \setminus V']$?

MID directly emerges from a problem concerning electoral control with respect to so-called "Llull voting" [2,9], one of the well-known voting systems based on pairwise comparision of candidates. Concerning MDD, in undirected social networks the degree of a vertex relates to its popularity or influence [18, pages 178–180]. Then, making a distinguished vertex minimum degree (equivalently, making it maximum degree in the complement graph) corresponds to activities or campaigns where a single agent shall be transformed to the least or most important agent in its community. Minimum vertex deletion, hence, can be interpreted as making "competing agents" disappear at minimum cost. A problem related to MDD is BOUNDED DEGREE DELETION (BDD) and its dual problem (considering the complement graph) MAXIMUM s-PLEX. For BDD the goal is to bound the maximum vertex degree by a prespecified value d (the case $d = 0$ is equivalent to the well-known VERTEX COVER problem) using a minimum number of vertex deletions. Other than MDD, BDD and its dual MAXIMUM s-PLEX have been studied quite intensively in recent years [1,10,14] which is due to their applications in social and biological network analysis.

Although both MID and MDD are simple and natural graph problems, we only know one previous publication concerning these problems. MID has been shown W[2]-complete for parameter solution size k even when restricted to tournament graphs and it is polynomial-time solvable on directed acyclic graphs [2].

We initiate a thorough theoretical analysis of MID and MDD mainly focussing on "tree-likeness" parameterizations. We employ several basic structural parameters measuring the tree-likeness of graphs. The most famous parameter is the *treewidth* tw of the input graph, which comes along with the concept of tree decompositions of graphs.[1] The *feedback vertex set number* s_v of a graph is the minimum number of vertices to delete from a graph to make it acyclic. Correspondingly, the *feedback edge set number* s_e and the the *feedback arc set number* s_a, respectively, denote the minimum number of edges or arcs to delete from an undirected or directed graph to make it acyclic. While the computation

[1] We omit any details because we will not need the formal definition in this work.

Table 1. Overview of the parameterized complexity of Min-Indegree Deletion and Min-Degree Deletion. The considered parameters are tw :="treewidth of the input graph", s_v :="size of a feedback vertex set", s_a :="size of a feedback arc set", s_v^* := "size of a feedback vertex set not containing w_c", s_e :="size of a feedback edge set", k :="number of vertices to delete", and d :="maximum degree". New results are in boldface. The remaining results are from [2].

parameter	Min-Indegree Deletion	Min-Degree Deletion
tw	—	**FPT, no polynomial kernel**
s_v	**W[2]-hard**	**FPT, no polynomial kernel**
s_v^*	**W[2]-hard**	$\mathbf{O((2s_v^* + 4)^{s_v^*} \cdot n^6)}$**, no polynomial kernel**
s_a, s_e	**W[2]-hard**	$\mathbf{O(2^{s_e} \cdot n^3)}$**, vertex-linear kernel**
k	W[2]-complete	**W[1]-hard**
d	FPT	FPT
(s_v, k)	$\mathbf{O(s_v \cdot (k+1)^{s_v} \cdot n^2)}$**, no polynomial kernel**	

of tw, s_v and s_a leads to NP-hard problems, s_e can be quickly determined by a spanning tree computation. Note that a small value of s_e means that the network is very sparse—however, there are several sparse social networks [13,16,17].

Table 1 summarizes our results. We extend the previous results for MID [2] by showing that MID is W[2]-hard even when parameterized by s_a whereas it turns fixed-parameter tractable for the combined parameter (k, s_v). Note that this also implies fixed-parameter tractability with respect to the combined parameter (k, s_a) since s_a is a *weaker* parameter than s_v in the sense that $s_v \leq s_a$. As to MDD, we show that it is NP-complete as well as W[1]-hard with respect to the parameter k, devising a parameterized many-one reduction from the Independent Set problem. In addition, we show that MDD is fixed-parameter tractable for each of the tree-likeness parameters treewidth, size s_v^* of a feedback vertex set not containing the distinguished vertex, and feedback edge set number s_e. Herein, our fixed-parameter tractability result for treewidth is of purely theoretical interest whereas the one for the feedback edge set number comes along with a $2s_e$-vertex problem kernel and a size-$O(2^{s_e})$ search tree. The result for s_v^* relies on dynamic programming and bears a combinatorial explosion of $O((2s_v^* + 4)^{s_v^*})$. Finally, building on a recent framework for proving non-existence of polynomial-size problem kernels [3], we also show that there is presumably no polynomial-size problem kernel for MDD even for the combined parameter (k, s_c^*), where s_c^* denotes the size of a vertex cover *not* containing the distinguished vertex. This directly implies the non-existence of polynomial-size problem kernels for the parameters feedback vertex set number and treewidth. Due to the lack of space, several details are deferred to a full version of this paper.

Preliminaries. Parameterized complexity is a two-dimensional framework for studying the computational complexity of problems [8,11,15]. One dimension is the input size n (as in classical complexity theory), and the other one is the *parameter* k (usually a positive integer). A problem is called *fixed-parameter tractable* (fpt) if it can be solved in $f(k) \cdot n^{O(1)}$ time, where f is a computable function only depending on k. The complexity class consisting of all fpt problems is denoted by FPT. A core tool in the development of fixed-parameter algorithms is polynomial-time preprocessing by *data reduction* [4,12]. Here, the goal is for a given problem instance x with parameter k, to transform it into a new instance x' with parameter k' such that the size of x' is upper-bounded by some function only depending on k, the instance (x, k) is a yes-instance if and only if (x', k') is a yes-instance, and $k' \leq k$. The reduced instance, which must be computable in polynomial time, is called a *problem kernel*, and the whole process is called *reduction to a problem kernel* or *kernelization*.

Downey and Fellows [8] developed a formal framework for showing *fixed-parameter intractability* by means of *parameterized reductions*. A parameterized reduction from a parameterized problem P to another parameterized problem P' is a function that, given an instance (x, k), computes in $f(k) \cdot n^{O(1)}$ time an instance (x', k') (with k' only depending on k) such that (x, k) is a yes-instance of problem P if and only if (x', k') is a yes-instance of problem P'. The basic complexity class for fixed-parameter intractability is called $W[1]$. There is good reason to believe that $W[1]$-hard problems are not fpt [8,11,15]. In this sense, $W[1]$-hardness is the parameterized complexity analog of NP-hardness. The class W[2] means the next higher degree of parameterized intractability.

We assume familiarity with basic graph-theoretic concepts. Let $G = (V, E)$ be an undirected graph. Unless stated otherwise, let $n := |V|$ and $m := |E|$. For $V' \subseteq V$ we denote the subgraph induced by V' as $G[V']$. Furthermore, we write $G - V'$ for $G[V \setminus V']$. The open neighborhood of a vertex v is denoted by $N(v)$ and the *degree* of v is $\deg(v) := |N(v)|$. We use analogous terms for directed graphs and differentiate between in- and out-(degree, neighborhood, etc.) by a subscript in the notation (e.g., $\deg_{\text{in}}(v)$ denotes the indegree of v).

2 Min-Degree Deletion

In this section, we study parameterizations of Min-Degree Deletion by the solution size, that is, the number of vertices to delete, and structural graph parameters measuring the tree-likeness. By devising a parameterized reduction from the W[1]-complete Independent Set problem we obtain the following.

Theorem 1. Min-Degree Deletion *is NP-complete and W[1]-hard for the parameter "number of vertices to delete".*

For the "tree-likeness" parameterizations, we show fixed-parameter tractability results (Subsection 2.1) and refute the existence of some polynomial-size problem kernels (Subsection 2.2).

2.1 Fixed-parameter tractability results

In the following, all structural graph parameters are related to measuring the tree-likeness of the underlying graph. More specifically, we provide results for the treewidth tw, the size s_v^* of a feedback vertex set not containing the distinguished vertex, and the feedback edge set number s_e. By definition, $(\text{tw}+1) \leq s_v^* \leq s_e$. Hence, our fixed-parameter tractability result for MDD for the parameter tw implies fixed-parameter tractability for the parameters s_v^* and s_e. However, since our corresponding result for tw only provides a classification and not an efficient fixed-parameter algorithm, we subsequently present specific fixed-parameter tractability results for each parameterization.

Parameter treewidth. For treewidth, the "strongest" tree-likeness parameter we consider in this section, we obtain the following.

Theorem 2. MIN-DEGREE DELETION *is fixed-parameter tractable for the parameter treewidth.*

The proof of Theorem 2 relies on expressing MDD by a monadic second-order logic (MSO) sentence and making use of Courcelle's famous theorem [6]. Due to the huge constants coming along with Courcelle's machinery this result is of purely theoretical interest. The following observation is crucial to obtain Theorem 2 as well as for some of our other results.

Observation 1 *Let $G = (V, E)$ be a graph of treewidth* tw *and let M^* be any solution set for* MDD. *Then, w_c has degree at most* $\text{tw}-1$ *in $G - M^*$.*

Parameter distinguished feedback vertex set number. We investigate the parameter *distinguished feedback vertex set number* s_v^* denoting the "size of a feedback vertex set not containing the distinguished vertex w_c". Since for a graph with treewidth tw it holds that $s_v^* \geq s_v \geq (\text{tw}+1)$, Theorem 2 implies that MDD is fixed-parameter tractable with respect to s_v^*. However, Theorem 2 does not come with a direct combinatorial algorithm and hence such an algorithm with running time $O((2s_v^*+4)^{s_v^*} \cdot n^4 \cdot \deg(w_c)^2)$ will be provided in the following.

Let $(G = (V, E), w_c, k)$ be an MDD-instance and let V_f be a feedback vertex set that does not contain w_c. Our algorithm basically branches into all possible subsets V_f^* of V_f and investigates whether there is a solution containing all vertices from V_f^* and not containing any vertex from $V_f \setminus V_f^*$. Furthermore, the algorithm iterates over the "final" degree that w_c might assume in the graph G after deleting a set of "solution vertices". Additionally applying some simple branching and preprocessing steps it remains to solve the following problem.

ANNOTATED MIN-DEGREE DELETION (AMDD)
Given: An undirected graph $G = (V, E)$, a distinguished vertex w_c, a feedback vertex set V_f of G with $V_f \subseteq V \setminus \{w_c\}$, and two non-negative integers k and i.
Question: Is there a subset $M \subseteq V \setminus (V_f \cup \{w_c\})$ of size at most k such that, in $G - M$, $\deg(w_c) = i$ and every other vertex has degree at least $i + 1$?

The branching and the preprocessing giving an AMDD-instance can be carried out in $O(2^{|V_f|} \cdot n^2)$ time. Moreover, due to the preprocessing, in the following we can assume that every vertex in $V \setminus \{w_c\}$ has degree at least $i+1$ and w_c has at most i neighbors in V_f. Now, for an AMDD-instance $(G = (V, E), w_c, V_f, k, i)$, the algorithm makes use of the following property of $V_S := V \setminus (V_f \cup \{w_c\})$, the set consisting of all vertices that can be part of the solution.

Observation 2 *Let $n_1, \ldots, n_d$ denote the neighbors of w_c in $G - V_f$. In the graph $G[V_S]$, every vertex n_x belongs to a connected component $T(x)$ such that $T(x)$ is a tree and does not contain any vertex n_y with $n_x \neq n_y$.*

Observation 2 can be seen as follows. Consider two neighbors n_x and n_y of w_c. First, assume that there would be a path from n_x and n_y that does not contain w_c. Adding w_c to this path would induce a cycle and hence V_f would not be a feedback vertex set of G. Hence, every connected component can contain at most one neighbor of w_c. Second, a cycle within a connected component would also violate that V_f is a feedback vertex set. Hence, all connected components induce trees.[2]

Now, we take a look at an arbitrary solution set M of our MDD-instance. Since the final degree of w_c is i, M must contain $\deg_G(w_c) - i$ neighbors of w_c. Putting a vertex $x \in N(w_c) \setminus V_f$ into the solution may decrease the degree of other vertices from $T(x)$ so that they also must be part of the solution. The set $A(x)$ of *affected* vertices that need to be deleted when x is deleted can be computed iteratively as follows. Start with $A(x) := \{x\}$. While there is vertex v with degree at most i in $T(x) - A(x)$, add v to $A(x)$. Since we have to put all vertices of $A(x)$ into a solution when choosing x into the solution, we define the *cost* of x as $\mathrm{cost}(x) := |A(x)|$. Moreover, we will make use of the following easy-to-verify observation.

Observation 3 *A vertex $v \in V_s \setminus (\bigcup_{x \in N(w_c) \setminus V_f} A(x))$ cannot be part of any minimal solution.*

For the graph not containing vertices from the feedback vertex set V_f, a solution could easily be computed by choosing a set of $\deg(w_c) - i$ neighbors of w_c such that the sum of the corresponding costs is minimal. The critical point is that putting a vertex x into the solution set may also decrease the degree of vertices from V_f. By definition, we cannot remove any vertex from V_f. Thus, we must ensure that we "keep" enough vertices from V_s such that the final degree of every vertex from V_f is at least $i + 1$. For every vertex $v \in V_f$, we can easily compute the number $n_{\text{fixed}}(v)$ of neighbors which it has "for sure" in every minimal solution. More specifically, $n_{\text{fixed}}(v)$ is the number of neighbors of v in V_f and in $V_s \setminus (\bigcup_{x \in N(w_c)} A(x))$ (see Observation 3).

We introduce some notation measuring how many neighbors of a vertex from V_f must be kept under the assumption that a certain subset $V_r \subseteq V_s$

[2] Observation 2 does not hold for a feedback vertex set containing the distinguished vertex. Hence, the following approach cannot be transferred to this more general case.

is *not* part of a solution. More specifically, for a vertex $v \in V_f$, let $n_{V_r}(v)$ be the number of neighbors of v in V_r. Then, the number of additional neighbors that are not allowed to be deleted is defined as $s(v, V_r) := i + 1 - n_{\text{fixed}}(v) - n_{V_r}(v)$. This can be generalized as follows.

Definition 1. *For $V_f = \{v_1, \ldots, v_{|V_f|}\}$, the* remain-tuple *with respect to $V_r \subseteq V_S$ is $S = (s_1, \ldots, s_{|V_f|})$ where $s_j := s(v_j, V_r)$.*

Recall that the task is to search for a set $N \subseteq N(w_c) \setminus V_f$ of $\deg(w_c) - i$ neighbors of w_c of minimum cost such that the degree of every vertex from V_f is at least $i + 1$. Now, for every subset $N' \subseteq N(w_c) \setminus V_f$, the effect of choosing that N' is *not* part of a solution can be described by a remain-tuple. More specifically, a subset $N' \subseteq N(w_c) \setminus V_f$ *realizes* a remain-tuple $(s'_1, \ldots, s'_{|V_f|})$ when, for every $v \in V_f$, the number of neighbors of v in $\bigcup_{x \in N'} A(x)$ is at least $i + 1 - n_{\text{fixed}}(v) - s'_i$. Then, a cost-$k$ set $N \subseteq N(w_c)$ containing $\deg(w_c) - i$ neighbors of w_c such that set $N(w_c) \setminus N$ realizes the remain-tuple $(0, \ldots, 0)$ corresponds to a solution.

Dynamic programming table. Based on the previous definitions, the dynamic programming table is defined as $D(x, z, S')$ with $x \in \{1, \ldots, d\}$ where $d := |N(w_c) \cap V_s|$, $z \leq \min\{x, d - i\}$, and $S' \subseteq \mathcal{S} := \{(s'_1, \ldots, s'_{|V_f|}) \mid 0 \leq s'_j \leq i + 1\}$. The entry $d(x, z, S')$ contains the minimum cost of deleting a size-z subset $N' \subseteq \{n_i \in N(w_c) \mid i \leq x\}$ such that $N'_r := N(w_c) \setminus N'$ "realizes" the remain-tuple S'. It follows that $D(\deg(w_c), \deg(w_c) - i, (0, \ldots, 0)) \leq k$ if and only if (G, V_f, w_c, k, i) is a yes-instance of AMDD. It is easy to verify that the size of the D is bounded by $\deg(w_c)^2 \cdot (s^*_v + 2)^{s^*_v}$ (see also Observation 1).

One can show that the initialization and update step per entry can be accomplished in $O((s^*_v + 2)^{s^*_v} \cdot n^2 \cdot \deg(w_c)^2)$ time. Hence, together with the running time for the overall branching into all subsets of a feedback vertex set, one ends up with the following.

Theorem 3. Min-Degree Deletion *can be solved in $O((2s^*_v + 4)^{s^*_v} \cdot n^4 \cdot \deg(w_c)^2)$ time with s^*_v being the size of a feedback vertex set not containing w_c.*

Parameter feedback edge set number. As discussed in the beginning of this section, the feedback edge set number is the weakest of our parameters measuring the tree-likeness of graphs. Hence, not surprisingly, we achieve our best running time bounds here, based on kernelization and a simple search tree.

Our problem kernel result relies on the following "low-degree removal" procedure. Let $G = (V, E)$ be an undirected graph and k be a positive integer. Denote by $\text{RLD}(G, k)$ the graph resulting from the following data reduction: If deleting all or all but one neighbors from w_c leads to a solution (by iteratively deleting all further vertices with degree at least zero/one), then return "yes". Otherwise, w_c has degree at least two for every solution. Hence, iteratively remove every vertex with degree at most two and decrease k accordingly. It is easy to verify that $\text{RLD}(G, k)$ is sound and can be executed in $O(n^2 \cdot k)$ time. Note that every vertex different from w_c in $\text{RLD}(G, k)$ has degree at least three.

Theorem 4. *Parameterized by the feedback edge set number s_e,* Min-Degree Deletion *admits a $2s_e$-vertex problem kernel which can be computed in $O(n^2 \cdot k)$ time.*

Proof. Let $(G', k') := \mathrm{RLD}(G, k)$, and let E_d be a size-s_e-feedback edge set for G. The graph $G - E_d$ is a forest. Since each vertex in G' has degree at least three, each leaf in $G' - E_d$ must be incident to at least two edges in E_d. It follows that $G' - E_d$ contains $l \leq s_e$ leaves because each leaf must be incident to two edges of the feedback edge set and each edge of the feedback edge set can be incident to at most two leaves. Furthermore, the number of incidences of the edges in E_d is bounded from above by $2s_e$. Each inner vertex of degree two in $G' - E_d$ must be incident to at least one edge in E_d. Since there are l leaves in $G' - E_d$, only $2s_e - 2l$ incidences are left over. Hence, $G' - E_d$ contains at most $2s_e - 2l$ inner vertices with degree two. Moreover, all remaining vertices must have degree at least three and a tree with l leaves can clearly have at most l such vertices. Altogether, G' consists of at most $l + 2s_e - 2l + l = 2s_e$ vertices. □

Finally, we complement Theorem 4 by a simple search tree algorithm which can be interleaved with the data reduction procedure $\mathrm{RLD}(G, k)$. This yields the following theorem.

Theorem 5. Min-Degree Deletion *can be solved in $O(2^{s_e} \cdot {s_e}^3 + n^2 \cdot k)$ time, where s_e is the feedback edge set number of the input graph.*

2.2 Non-existence of a polynomial kernel

We show that, unless coNP $\subseteq$ NP / poly, there is no polynomial kernel for MDD with respect to the parameter $s_c^* :=$"size of a vertex cover that does not contain w_c". Since the treewidth t_w and the feedback vertex set number s_v of a graph are bounded from above by s_c^*, this non-kernelizable result carries over to these two parameterizations.

Theorem 6. MDD *does not admit a polynomial kernel with respect to the combined parameter (s_c^*, k), with s_c^* being the size of a vertex cover not containing w_c and k being the solution size, unless* coNP $\subseteq$ NP / poly.

Proof. Our proof relies on a reduction from Hitting Set (HS) defined as follows. Given a set family $\mathcal{S} := \{S_1^*, \ldots, S_m^*\}$ over a universe $U := \{u_1^*, \ldots, u_d^*\}$ and an integer $k' \geq 0$, HS asks for a subset $U' \subseteq U$ with $|U'| \leq k'$ such that $S_i^* \cap U' \neq \emptyset$ for every i, $1 \leq i \leq m$. Herein, U' is called a *hitting set.*

Dom et al. [7] have shown that HS does not admit a problem kernel of size $(d+k')^{O(1)}$, unless coNP $\subseteq$ NP / poly. Since HS and MDD are NP-complete, it directly follows from a result of Bodlaender et al. [5] that if there is a polynomial-time reduction from HS to MDD such that $(s_c^* + k) \leq (d+k')^{O(1)}$, then MDD does not admit a polynomial kernel with respect to (s_c^*, k) unless coNP $\subseteq$ NP / poly. In the following, we provide such a reduction.

Let (U, S, k') be an HS-instance. We construct an undirected graph $G = (V, E)$ with a distinguished vertex w_c as follows. The vertex set V is the disjoint union of the sets $\{w_c\}$, V_U, $V_{\mathcal{S}}$, C, and L. Herein, $V_U := \{u_i \mid u_i^* \in U\}$, $V_{\mathcal{S}} := \{s_j \mid S_j^* \in \mathcal{S}\}$, $C := \{c_1, \ldots, c_{k'+1}\}$, and $L := \{l_1, \ldots, l_d\}$. The edge set is constructed as follows. There is an edge between u_i and s_j if and only if $u_i^* \in S_j^*$. Moreover, the following edges are added to adjust the degree of w_c to d and, for each other vertex, to at least $k'+1$. First, w_c is made adjacent to every vertex in V_U. Furthermore, C is transformed into a clique, and each l_i, $1 \leq i \leq d$, is made adjacent to each vertex in C. Finally, each vertex $x \in V_U \cup V_{\mathcal{S}}$ is made adjacent to k' arbitrarily chosen vertices of C. This completes the construction.

Now, observe that each edge of G is incident to a vertex in $C \cup V_U$. Hence, G has a vertex cover of size $k'+1+d$ which does not contain w_c. For the correctness of the reduction it remains to show that $(U, \mathcal{S}, k')$ is a yes–instance of HS if and only if $(G, w_c, d-k')$ is a yes–instance of MDD.

"$\Rightarrow$": Let $U' \subseteq U$ with $|U'| = k'$ denote a hitting set of $\mathcal{S}$. We show that $M := \{u_j \mid u_j^* \in U \setminus U'\}$ is a solution for $(G, w_c, d-k')$. First, observe that w_c has degree k' in $G - M$. Moreover, since U' is a hitting set, every vertex in $V_{\mathcal{S}}$ has at least one neighbor in $V_U \setminus M$, and, hence, degree at least $k'+1$ in $G-M$. For this reason and since we do not delete a neighbor of $L \cup C$, each vertex in $V \setminus \{w_c\}$ has degree at least $k'+1$. Hence, $(G, w_c, d-k')$ is a yes-instance of MDD.

"$\Leftarrow$": Let $M \subseteq V$ with $|M| \leq d-k'$ denote a solution for $(G, w_c, d-k')$. First, we argue that w_c has degree k' in $G-M$. Clearly, w_c cannot have degree smaller than k'. Furthermore, w_c cannot have degree more than k' in $G-M$; otherwise, since w_c is the only vertex with minimum degree in $G-M$ and each vertex in L has degree $k'+1$, M must contain every vertex in L. However, $|L| = d > d-k'$. Thus, $\deg_{G-M}(w_c) = k'$ and, as a consequence, $M \subseteq V_U$ and $|M| = d-k'$.

Next, we show that $U' := \{u_i^* \in U \mid u_i \in V_U \setminus M\}$ is a hitting set of size k'. By the observation above, $|U'| = k'$. Assume towards a contradiction that there is a set S_j^*, $1 \leq j \leq m$, with $S_j^* \cap U' = \emptyset$. Thus, for each element $u_i^* \in S_j^*$ the corresponding vertex u_i is in M. Due to the construction of G, vertex s_j has degree k' in $G-M$; since $\deg_{G-M}(w_c) = k'$ this contradicts the fact that w_c is the only vertex with minimum degree. □

Since the treewidth and the feedback vertex set of a graph are bounded from above by s_c^*, we arrive at the following.

Corollary 1. MDD *has no polynomial problem kernel with respect to the parameters feedback vertex set and treewidth, respectively, unless* coNP $\subseteq$ NP / poly.

3 Min-Indegree Deletion

In this section, we show that MID is W[2]-hard with respect to the parameter feedback arc set number s_a. We provide a parameterized reduction from the W[2]-complete Dominating Set (DS) problem [8]. Given an undirected graph and an integer k, DS asks whether there is a size-k subset $V' \subseteq V$ such that every vertex from V is contained in V' or has a neighbor in V'. A corresponding subset is denoted as *dominating set*.

Theorem 7. Min-Indegree Deletion *is W[2]-hard with respect to the feedback arc set number s_a.*

Proof. Given a DS-instance $(G^* = (V^*, E^*), k)$ with $V^* = \{v_1^*, v_2^*, \dots, v_n^*\}$, we construct a directed graph $G = (W, E)$ with feedback arc set number at most $(k+1)^2$ such that $(G, w_c, n-k)$ is a yes-instance of MID if and only if (G^*, k) is a yes-instance of DS.

The vertex set W of G consists of w_c and the union of the following disjoint vertex sets. The sets $V := \{v_i \mid v_i^* \in V^*\}$ and $D := \{d_i \mid v_i^* \in V^*\}$ representing the vertices of G and four sets of auxiliary vertices, namely a set S containing n vertices and three sets X, Y, and Z, each containing $k+1$ vertices. The arcs of G are as follows.

- One arc from v_i to d_j if and only if $v_j^* \in N[v_i^*]$.
- One arc from each vertex in V to w_c.
- One arc from each vertex in X to each vertex in Y, from each vertex in Y to each vertex in Z, and from each vertex in Z to each vertex in X.
- One arc from each of k arbitrarily chosen vertices in Y to each vertex in D.
- One arc from each vertex in Y to each vertex in V.
- One arc from each vertex in X to each vertex in S.

It follows directly from the construction that the distinguished vertex w_c has indegree n and each vertex in $V \cup X \cup Y \cup Z \cup S$ has indegree $k+1$. Since each vertex d_i has one ingoing arc from v_i and k ingoing neighbors from Y, the vertices in D have indegree at least $k+1$.

Furthermore, it is easy to verify that $(W, E \setminus (X \times Y))$ is acyclic and, since $|X| = |Y| = k+1$, the feedback arc set number s_a is at most $(k+1)^2$. This finishes the description of the construction. It remains to prove the correctness.

Claim: (G^*, k) is a yes-instance of DS if and only if $(G, w_c, n-k)$ is a yes-instance of MID.

"$\Rightarrow$": Let $V_d^* \subseteq V^*$ be a size-k dominating of G^*. We show that $M_d := \{v_i \in V \mid v_i^* \notin V_d^*\}$ is a solution for MID. Since $|M_d| = n-k$ and w_c has indegree n in G, w_c has indegree k in $G - M_d$. We show that all other vertices have degree at least $k+1$. By construction, every vertex in G has indegree at least $k+1$. Since from the vertices in V_d^* there are only arcs to $D \cup \{w_c\}$, only vertices from $D \cup \{w_c\}$ can have smaller indegrees in $G - M_d$ than in G. Because V_d^* is a dominating set, every d_i has at least one in-neighbor within $V \setminus M_d$. Moreover, every d_i has k further in-neighbors in Y. Hence, each vertex in D has indegree at least $k+1$. Thus, $(G, w_c, n-k)$ is a yes-instance of MID.

"$\Leftarrow$": Consider a yes-instance $(G, w_c, n-k)$ of MID with solution M_d. We show that $V_d^* := \{v_i^* \in V^* \mid v_i \in V \setminus M_d\}$ is a size-k dominating set of G^*.

We first prove that V_d^* has cardinality k. To this end, we show by contradiction that the indegree of w_c in $G - M_d$ is k and hence M_d contains only vertices from V. Assume that w_c has indegree at least $k+1$ in $G - M_d$. Then, every other vertex must have indegree greater than $k+1$ in $G - M_d$. Since every vertex in S has indegree exactly $k+1$, it follows that $S \subseteq M_d$ and hence $|M_d| \geq n$; a contradiction. Consequently, $|V \cap M_d| = n-k$ and, hence, V_d^* has cardinality k.

It remains to show that V_d^* is a dominating set. Assume that there is a vertex $v_i^* \in V^*$ not dominated by any vertex in V_d^*. This implies that d_i has no in-neighbor from V in $G - M_d$. Moreover, by construction, d_i has only k in-neighbors in $G - V$. As argued above, d_i is not in M_d since M_d contains only vertices from V. Hence, d_i and w_c have indegree k in $G - M_d$; a contradiction.

In the remainder of this section, we show fixed-parameter tractability of MID with respect to the combined parameter feedback vertex set number s_v and number k of vertices to be deleted. The corresponding branching algorithm relies on the following lemma.

Lemma 1. *For a yes-instance* $(G = (V, E), w_c, k)$ *of* MID*, the indegree of* w_c *in* G *is at most* $k + s_v$*, where* s_v *denotes the feedback vertex set number of* G*.*

Proof. The proof is by contradiction. Let $V_f \subseteq V$ be a feedback vertex set of size s_v. Assume that $\deg_{\text{in}}(w_c) > s_v + k$. For every subgraph G' of G obtained by deleting k vertices from $G - \{w_c\}$, one can make the following two observations. First, since $G' - V_f$ is acyclic, there must be a vertex v with indegree zero in $G' - V_f$. Hence, the indegree of v in G' is at most s_v (in case that v has one ingoing arc from every vertex in V_f). Second, since $\deg_{\text{in}}(w_c) > s_v + k$ in G, it follows that $\deg_{\text{in}}(w_c) > s_v$ in G'. Consequently, there is no size-k subset of vertices that can be deleted from G such that w_c is a vertex with minimum indegree; a contradiction to the fact that $(G = (V, E), w_c, k)$ is a yes-instance. □

Now, by applying an algorithm branching on all up-to-size-k subsets of the in-neighborhood of w_c and checking whether a corresponding subset can be extended to a solution, one arrives at the following theorem.

Theorem 8. Min-Indegree Deletion *can be solved in* $O((k+1)^{s_v} \cdot s_v \cdot n^2)$ *time.*

4 Conclusion

We introduced the NP-hard vertex deletion problem Min-Degree Deletion on undirected graphs. For Min-Degree Deletion and its directed counterpart Min-Indegree Deletion we provided several results concerning their fixed-parameter tractability with respect to the parameter solution size and several parameters measuring the input graph's tree-likeness (see Table 1 in the introductory section for an overview). There remain numerous opportunities for future research. For example, the fixed-parameter tractability results for Min-Degree Deletion for the parameter treewidth as well as for the parameter feedback vertex set are far from any practical relevance. For these parameterizations it would be interesting to complement our classification results by direct combinatorial algorithms. Moreover, we are not aware of studies concerning the polynomial-time approximability of both problems.

References

1. B. Balasundaram, S. Butenko, and I. V. Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Oper. Res.*, 2010. To Appear.
2. N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009.
3. H. Bodlaender, R. Downey, M. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. System Sci.*, 75(8):423–434, 2009.
4. H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proc. 4th IWPEC*, volume 5917 of *LNCS*, pages 17–37. Springer, 2009.
5. H. L. Bodlaender, S. Thomassé, and A. Yeo. Analysis of data reduction: Transformations give evidence for non-existence of polynomial kernels. Technical Report UU-CS-2008-030, Department of Information and Computing Sciences, Utrecht University, 2008.
6. B. Courcelle. Graph rewriting: An algebraic and logic approach. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 193–242. 1990.
7. M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In *Proc. 36th ICALP I*, volume 5555 of *LNCS*, pages 378–389. Springer, 2009.
8. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.
9. P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *J. Artifical Intelligence Res.*, 35(1):275–341, 2009.
10. M. R. Fellows, J. Guo, H. Moser, and R. Niedermeier. A generalization of Nemhauser and Trotter's local optimization theorem. In *Proc. 26th STACS*, pages 409–420. IBFI Dagstuhl, Germany, 2009.
11. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.
12. J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007.
13. B. A. Huberman, D. M. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 14(1), 2009.
14. H. Moser, R. Niedermeier, and M. Sorge. Algorithms and experiments for clique relaxations—finding maximum s-plexes. In *Proc. 8th SEA*, volume 5526 of *LNCS*, pages 233–244. Springer, 2009.
15. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
16. J. Potterat, L. Phillips-Plummer, S. Muth, R. Rothenberg, D. Woodhouse, T. Maldonado-Long, H. Zimmerman, and J. Muth. Risk network structure in the early epidemic phase of HIV transmission in Colorado Springs. *Sexually Transmitted Infections*, 78 (Supplement 1):159–163, 2002.
17. C. Romm-Livermore and K. Setzekorn. *Social Networking Communities and E-Dating Services: Concepts and Implications.* Information Science Reference, 2008.
18. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications.* Cambridge University Press, 1994.