SIAM J. DISCRETE MATH.
Vol. 29, No. 1, pp. 122–144



# FIXED-PARAMETER TRACTABILITY OF MULTICUT IN DIRECTED ACYCLIC GRAPHS*

STEFAN KRATSCH†, MARCIN PILIPCZUK‡, MICHAŁ PILIPCZUK§, AND MAGNUS WAHLSTRÖM¶

**Abstract.** The MULTICUT problem, given a graph $G$, a set of terminal pairs $\mathcal{T} = \{(s_i, t_i) \mid 1 \leq i \leq r\}$, and an integer $p$, asks whether one can find a *cutset* consisting of at most $p$ nonterminal vertices that separates all the terminal pairs, i.e., after removing the cutset, $t_i$ is not reachable from $s_i$ for each $1 \leq i \leq r$. The fixed-parameter tractability of MULTICUT in undirected graphs, parameterized by the size of the cutset only, has been recently proved by Marx and Razgon [*SIAM J. Comput.*, 43 (2014), pp. 355–388] and, independently, by Bousquet, Daligault, and Thomassé [*Proceedings of STOC*, ACM, 2011, pp. 459–468], after resisting attacks as a long-standing open problem. In this paper we prove that MULTICUT is fixed-parameter tractable on directed acyclic graphs when parameterized both by the size of the cutset and the number of terminal pairs. We complement this result by showing that this is implausible for parameterization by the size of the cutset only, as this version of the problem remains $W[1]$-hard.

**Key words.** fixed-parameter tractability, multicut, directed acyclic graph, important separators, shadow removal

**AMS subject classifications.** 05C85, 68R10, 68W05

**DOI.** 10.1137/120904202

## 1. Introduction.

Parameterized complexity is an approach for tackling hard problems by designing algorithms that perform robustly, when the input instance is in some sense simple; its difficulty is measured by an integer that is additionally appended to the input, called the *parameter*. Formally, we say that a problem is *fixed-parameter tractable* (FPT) if it can be solved by an algorithm that runs in time $f(k)n^c$ for $n$ being the length of the input and $k$ being the parameter, where $f$ is some computable function and $c$ is a constant independent of the parameter.

The search for fixed-parameter algorithms resulted in the introduction of a number of new algorithmic techniques and gave fresh insight into the structure of many classes of problems. One family that has received a lot of attention recently is the so-called graph cut problems, where the goal is to make the graph satisfy a global separation requirement by deleting as few edges or vertices as possible (depending on

*Received by the editors January 2, 2013; accepted for publication (in revised form) October 21, 2014; published electronically January 15, 2015. A preliminary version of this paper was presented at the 39th International Colloquium on Automata, Languages and Programming (ICALP 2012).
http://www.siam.org/journals/sidma/29-1/90420.html

†Technische Universität Berlin, 10623 Berlin, Germany (stefan.kratsch@tu-berlin.de). This research was done while the author was at the University of Utrecht, Netherlands, and was supported by the Netherlands Organization for Scientific Research (NWO), project "KERNELS: Combinatorial Analysis of Data Reduction."

‡University of Warwick, Coventry CV4 7AL, UK (malcin@mimuw.edu.pl). This research was done while the author was at the University of Warsaw, Poland, and was supported by NCN grant N206567140 and the Foundation for Polish Science.

§University of Warsaw, Krakowskie Przedmiescie 26/28, 00-927 Warszawa, Poland (michal.pilipczuk@mimuw.edu.pl). This research was done while the author was at the University of Bergen, Norway, and was supported by European Research Council (ERC) grant "Rigorous Theory of Preprocessing," reference 267959.

¶Royal Holloway University of London, Egham Hill, Egham, Surrey TW20 0EX, UK (Magnus.Wahlstrom@rhul.ac.uk). This research done while the author was at the Max-Planck Institute for Informatics, Saarbrücken, Germany.

the variant). Graph cut problems in the context of fixed-parameter tractability were to our knowledge first introduced explicitly in the seminal work of Marx [12], where it was proved that (i) MULTIWAY CUT (separate all terminals from each other by a cutset of size at most $p$) in undirected graphs is FPT when parameterized by the size of the cutset; (ii) MULTICUT in undirected graphs is FPT when parameterized by both the size of the cutset and the number of terminal pairs. Fixed-parameter tractability of MULTICUT parameterized by the size of the cutset only was left open by Marx [12] and resolved only much later (see below).

Probably the most fruitful contribution of the work of Marx [12] is the concept of *important separators*, which proved to be a tool almost perfectly suited to capturing the bounded-in-parameter character of sensible cutsets. The technique proved to be extremely robust and serves as the key ingredient in a number of FPT algorithms [13, 12, 4, 8, 14, 3, 10, 11, 6]. In particular, the fixed-parameter tractability of SKEW MULTICUT in directed acyclic graphs (DAGs), obtained via a simple application of important separators, enabled the first FPT algorithm for DIRECTED FEEDBACK VERTEX SET [4], resolving another long-standing open problem.

However, important separators have a drawback in that not all graph cut problems admit solutions with "sensible" cutsets in the required sense. This is particularly true in directed graphs, where, with the exception of the aforementioned SKEW MULTICUT problem in DAGs, for a long time few FPT graph cut problems were known; in fact, up until very recently it was open whether MULTIWAY CUT in directed graphs admits an FPT algorithm even in the restricted case of two terminals. The same complication arises in the undirected MULTICUT problem parameterized by the size of the cutset.

After a long struggle, MULTICUT was shown to be FPT by Marx and Razgon [13] and, independently, by Bousquet, Daligault, and Thomassé [2]. The key component in the algorithm of Marx and Razgon [13] is the technique of *shadow removal*, which, in some sense, serves to make the solutions to cut problems more well-behaved. This was adapted to the directed case by Chitnis, Hajiaghayi, and Marx [6], who proved that MULTIWAY CUT, parameterized by the size of the cutset, is FPT for an arbitrary number of terminals, by a simple and elegant application of the shadow removal technique. A subsequent work by Chitnis et al. [5] improves the complexity bounds of the shadow removal technique and applies it to show fixed-parameter tractability of DIRECTED SUBSET FEEDBACK VERTEX SET. This gives hope that, in general, shadow removal may be helpful for the application of important separators to the directed world.

As for the directed MULTICUT problem, it was shown by Marx and Razgon [13] to be $W[1]$-hard when parameterized only by the size of the cutset but otherwise had unknown status, even for a constant number of terminals in a DAG. We note that this case is known to be NP-hard and APX-hard [1].

*Our results.* The main result of this paper is the proof of fixed-parameter tractability of the MULTICUT IN DAGs problem, formally defined as follows:

| MULTICUT IN DAGs | **Parameter:** $p + r$ |
|---|---|
| **Input:** DAG $G$, set of terminal pairs $\mathcal{T} = \{(s_i, t_i) \mid 1 \leq i \leq r\}$, $s_i, t_i \in V(G)$ for $1 \leq i \leq r$, and an integer $p$. | |
| **Question:** Does there exist a set $Z$ of at most $p$ nonterminal vertices of $G$, such that for any $1 \leq i \leq r$ the terminal $t_i$ is not reachable from $s_i$ in $G \setminus Z$? | |

More precisely, we prove the following theorem.

THEOREM 1.1. MULTICUT IN DAGs *can be solved in* $O^*(2^{O(rp(r+p^2))})$ *time.*

Note that throughout the paper we use $O^*$-notation to suppress polynomial factors. Note also that we focus on vertex cuts; it is well known that in the directed acyclic setting the arc- and vertex-deletion variants are equivalent (cf. [6]).

Our algorithm makes use of the shadow removal technique introduced by Marx and Razgon [13], adjusted to the directed setting by Chitnis, Hajiaghayi, and Marx [6] with the improved bounds of [5], as well as the basic important separators toolbox that can be found in [6]. We remark that the shadow removal is but one of a number of ingredients of our approach: in essence, the algorithm combines the shadow removal technique with a degree reduction for the sources in order to carefully prepare the structure of the instance for a simplifying branching step. For a more detailed overview of a single step of the algorithm we also refer to Figure 1 later in this work.

We complement the main result with two lower bounds. First, we show that the dependency on $r$ in the exponent is probably unavoidable.

THEOREM 1.2. MULTICUT IN DAGS*, parameterized by the size of the cutset only, is* $W[1]$*-hard.*

Thus, we complete the picture of parameterized complexity of MULTICUT IN DAGS. We hope that it is a step toward fully understanding the parameterized complexity of MULTICUT in general directed graphs.

As a related result, we establish NP-completeness of SKEW MULTICUT, which is the special case of MULTICUT IN DAGS where we are given $d$ sources $(s_i)_{i=1}^d$ and $d$ sinks $(t_i)_{i=1}^d$, and the set of terminal pairs is defined as $\mathcal{T} = \{(s_i, t_j) : 1 \leq i \leq j \leq d\}$. The significance of this problem is mainly that it is used as a core subroutine in the FPT algorithm for DIRECTED FEEDBACK VERTEX SET of Chen et al. [4]. A polynomial-time algorithm for SKEW MULTICUT might have given new insights into DIRECTED FEEDBACK VERTEX SET, with potential consequences both for the running time of the FPT algorithm and, arguably, for the existence of a polynomial kernel (which is now one of the major open problems in kernelization; see, e.g., [9]).

THEOREM 1.3. SKEW MULTICUT *is NP-complete even in the restricted case of two sinks and two sources.*

*Organization of the paper.* In section 2 we introduce some notation and recall the framework of important separators and shadow removal. Section 3 contains our algorithm for MULTICUT IN DAGS and the proof of Theorem 1.1, whereas section 4 contains the proofs of Theorems 1.2 and 1.3. We conclude with a discussion on open problems in section 5.

**2. Preliminaries.** For a directed graph $G$, by $V(G)$ and $E(G)$ we denote its vertex and arc set, respectively. For a vertex $v \in V(G)$, we define its *in-neighborhood* $N_G^-(v) = \{u : (u, v) \in E(G)\}$ and *out-neighborhood* $N_G^+(v) = \{u : (v, u) \in E(G)\}$; these definitions are extended to sets $X \subseteq V(G)$ by $N_G^-(X) = (\bigcup_{v \in X} N_G^-(v)) \setminus X$ and $N_G^+(X) = (\bigcup_{v \in X} N_G^+(v)) \setminus X$. The *in-degree* and *out-degree* of $v$ are defined as $|N_G^-(v)|$ and $|N_G^+(v)|$, respectively. In this paper we consider simple directed graphs only; if at any point a modification of the graph results in a multiple arc, we delete all copies of the arc except for one. By $G^{\text{rev}}$ we denote the graph $G$ with all the arcs reversed, i.e., $G^{\text{rev}} = (V(G), \{(v, u) : (u, v) \in E(G)\})$.

A *path* in $G$ is a sequence of pairwise distinct vertices $P = (v_1, v_2, \ldots, v_d)$ such that $(v_i, v_{i+1}) \in E(G)$ for any $1 \leq i < d$. If $v_1$ is the first vertex of the path $P$ and $v_d$ is the last vertex, we say that $P$ is a *$v_1v_d$-path*. We extend this notion to sets of vertices: if $v_1 \in X$ and $v_d \in Y$ for some $X, Y \subseteq V(G)$, then $P$ is an *$XY$-path* as well. For a path $P = (v_1, v_2, v_3, \ldots, v_d)$ the vertices $v_2, v_3, \ldots, v_{d-1}$ are the *internal vertices* of $P$. The set of internal vertices of a path $P$ is the *interior* of $P$. We say

that a vertex $v$ is *reachable* from a vertex $u$ in $G$ if there exists a $uv$-path in $G$. As the considered digraphs are simple, each path $P = (v_1, v_2, \ldots, v_d)$ has a unique *first arc* $(v_1, v_2)$ and a unique *last arc* $(v_{d-1}, v_d)$.

Let $(G, \mathcal{T}, p)$ be a MULTICUT IN DAGS instance with a set of $r$ terminal pairs $\mathcal{T} = \{(s_i, t_i) : 1 \leq i \leq r\}$. We call the terminals $s_i$ *source terminals* and the terminals $t_i$ *sink terminals*. We let $T^s = \{s_i : 1 \leq i \leq r\}$, $T^t = \{t_i : 1 \leq i \leq r\}$ and $T = T^s \cup T^t$.

Fix a topological order $\preceq$ of $G$. For any sets $X, Y \subseteq V(G)$, we may order the vertices of $X$ and $Y$ with respect to $\preceq$ and compare $X$ and $Y$ lexicographically; we refer to this order on subsets of $V(G)$ as the *lexicographical order*.

A set $Z \subseteq V(G)$ is called a *multicut* in $(G, \mathcal{T}, p)$ if $Z$ contains no terminals, but for each $1 \leq i \leq r$, $t_i$ is not reachable from $s_i$ in $G \setminus Z$. Given a MULTICUT IN DAGS instance $\mathcal{I} = (G, \mathcal{T}, p)$ a multicut $Z$ is called a *solution* if $|Z| \leq p$. A solution $Z$ is called a *lex-min solution* if $Z$ is lexicographically minimum solution in $\mathcal{I}$ among solutions of minimum possible size.

Consider the following easy reduction.

LEMMA 2.1. *There exists a polynomial-time algorithm that, given a* MULTICUT IN DAGS *instance* $(G, \mathcal{T}, p)$*, computes an equivalent instance* $(G', \mathcal{T}', p')$ *such that*

1. $|\mathcal{T}| = |\mathcal{T}'|$ *and* $p = p'$*;*
2. $\mathcal{T}' = \{(s'_i, t'_i) : 1 \leq i \leq r\}$ *and all terminals* $s'_i$ *and* $t'_i$ *are pairwise distinct;*
3. *for each* $1 \leq i \leq r$ *we have* $N^-_{G'}(s'_i) = 0$ *and* $N^+_{G'}(t'_i) = 0$*.*

*Proof.* To construct the graph $G'$, start with the graph $G$ and for each terminal $v$ of $\mathcal{T}$ replace $v$ with $p+1$ copies of $v$ (i.e., vertices with the same in- and out-neighborhoods as $v$; these copies are not terminals in $\mathcal{T}'$). Moreover, for each $1 \leq i \leq r$, add to $G'$ a new terminal $s'_i$ with arcs $\{(s'_i, u) : u \in N^+_G(s_i)\}$ and a new terminal $t'_i$ with arcs $\{(u, t'_i) : u \in N^-_G(t_i)\}$. Finally, take $\mathcal{T}' = \{(s'_i, t'_i) : 1 \leq i \leq r\}$ and $p' = p$.

To see the equivalence, first take a multicut $Z$ in $(G, \mathcal{T}, p)$ and an arbitrary $s'_i t'_i$-path $P' = (s'_i, v'_1, v'_2, \ldots, v'_d, t'_i)$ in $G'$. Assume without loss of generality (w.l.o.g.) that $P'$ visits at most one copy of each terminal $v$ in $\mathcal{T}$, or else $P'$ contains a shorter $s'_i t'_i$-path as a strict subpath. The path $P'$ induces a path $P = (s_i, v_1, v_2, \ldots, v_d, t_i)$ in $G$: $v_k = v'_k$ if $v'_k$ is present in $G$, and $v_k = v$ if $v'_k$ is one of the $p+1$ copies of a terminal $v$ in $\mathcal{T}$. The path $P$ is intersected by $Z$ on some nonterminal vertex $v_k$; as $v_k = v'_k$, $Z$ intersects $P'$. We infer that $Z$ is a multicut in $(G', \mathcal{T}', p')$ as well.

In the other direction, let $Z$ be a multicut in $(G', \mathcal{T}', p')$ of size at most $p$ and let $P = (s_i, v_1, v_2, \ldots, v_d, t_i)$ be an arbitrary $s_i t_i$-path in $G$. Construct a path $P' = (s'_i, v'_1, v'_2, \ldots, v'_d, t'_i)$ by taking $v'_k = v_k$ if $v_k$ is a nonterminal in $(G, \mathcal{T}, p)$ and taking $v'_k$ to be one of the copies of $v_k$ that is not contained in $Z$ otherwise; note that at least one copy is not contained in $Z$, as $|Z| \leq p$. As $Z$ is a multicut in $(G', \mathcal{T}', p')$, $Z$ intersects $P'$. By the choice of the internal vertices of $P'$, $Z$ intersects $P'$ on some vertex $v'_k = v_k$ for $v_k$ being a nonterminal in $(G, \mathcal{T}, p)$. Therefore $Z \cap V(G)$ is a multicut in $(G, \mathcal{T}, p)$. ∎

In our algorithm, the set of terminal pairs $\mathcal{T}$ is never modified, and we never add an arc into a source or out of a sink. Thus, we may assume that during the course of our algorithm all terminals are pairwise distinct and that $N^-_G(s_i) = N^+_G(t_i) = \emptyset$ for all $1 \leq i \leq r$.

For $v \in V(G)$, by $S(G, v)$ we denote set of source terminals $s_i$ for which there exists an $s_i v$-path in $G$. For a set $S \subseteq T^s$ by $V(G, S)$ we denote the set of nonterminal vertices $v$ for which $S(G, v) = S$.

**2.1. Important separators and shadows.** In the rest of this section we recall the notion of important separators by Marx [12], adjusted to the directed case by

Chitnis, Hajiaghayi, and Marx [6], as well as the shadow removal technique of Marx and Razgon [13] and Chitnis, Hajiaghayi, and Marx [6, 5].

DEFINITION 2.2 (separator [6, Definition 2.2]). *Let $G$ be a directed graph with terminals $T \subseteq V(G)$. Given two disjoint nonempty sets $X, Y \subseteq V(G)$, we call a set $Z \subseteq V(G)$ an $X - Y$* separator *if* (i) $Z \cap T = \emptyset$, (ii) $Z \cap (X \cup Y) = \emptyset$, (iii) *there is no path from $X$ to $Y$ in $G \setminus Z$. An $X - Y$ separator $Z$ is called* minimal *if no proper subset of $Z$ is an $X - Y$ separator.*

By $\mathrm{cut}_G(X, Y)$ we denote the size of a minimum $X-Y$ separator in $G$; $\mathrm{cut}_G(X, Y) = \infty$ if $G$ contains an arc going directly from $X$ to $Y$ (recall that in our instances all terminals have in- or out-degree zero). By Menger's theorem, $\mathrm{cut}_G(X, Y)$ equals the maximum possible size of a family of $XY$-paths with pairwise disjoint interiors.

DEFINITION 2.3 (important separator [6, Definition 4.1]). *Let $G$ be a directed graph with terminals $T \subseteq V(G)$ and let $X, Y \subseteq V(G)$ be two disjoint nonempty sets. Let $Z$ and $Z'$ be two $X - Y$ separators. We say that $Z'$ is* behind $Z$ *if any vertex reachable from $X$ in $G \setminus Z$ is also reachable from $X$ in $G \setminus Z'$. A minimal $X-Y$ separator $Z$ is an* important separator *if no other $X-Y$ separator $Z'$ satisfies $|Z'| \leq |Z|$ while being also behind $Z$.*

We also need some known properties of minimum size cuts (cf. [3, 6]).

LEMMA 2.4 ([6, Lemma B.4]). *Let $G$ be a directed graph with terminals $T \subseteq V(G)$. For two disjoint nonempty sets $X, Y \subseteq V(G)$, there exists exactly one minimum size important $X - Y$ separator.*

DEFINITION 2.5 (closest mincut). *Let $G$ be a directed graph with terminals $T \subseteq V(G)$. For two disjoint nonempty sets $X, Y \subseteq V(G)$, the unique minimum size important $X - Y$ separator is called the $X - Y$* mincut closest to $Y$. *The $X - Y$* mincut closest to $X$ *is the $Y - X$ mincut closest to $X$ in $G^{rev}$.*

LEMMA 2.6. *Let $G$ be a directed graph with terminals $T \subseteq V(G)$ and let $X, Y \subseteq V(G)$ be two disjoint nonempty sets. Let $B$ be the unique minimum size important $X-Y$ separator, that is, the $X-Y$ mincut closest to $Y$, and let $v \in B$ be an arbitrary vertex. Construct a graph $G'$ from $G$ as follows: delete $v$ from $G$ and add an arc $(x, w)$ for each $x \in X$ and $w \in N^+_G(v) \setminus X'$, where $X'$ is the set of vertices reachable from $X$ in $G \setminus B$. Then the size of any $X - Y$ separator in $G'$ is strictly larger than $|B|$.*

*Proof.* The claim is obvious if $Y \cap N^+_G(v) \neq \emptyset$, as then $G'$ contains a direct arc from $X$ to $Y$. Therefore, let us assume that $Y \cap N^+_G(v) = \emptyset$.

We prove the lemma by exhibiting more than $|B|$ $XY$-paths in $G'$ that have pairwise disjoint interiors. Recall that $X' \supseteq X$ is the set of vertices reachable from $X$ in $G \setminus B$; note that $N^+_G(X') = B$. Since $B$ is a minimum size $X - Y$ separator, there exist a set of $XY$-paths $(P_u)_{u \in B}$ such that $P_u$ intersects $B$ only in $u$ and the interiors of paths $P_u$ are pairwise disjoint. Note that each path $P_u$ can be split into two parts: $P^X_u$, between $X$ and $u$, with all vertices except for $u$ contained in $X'$, and $P^Y_u$, between $u$ and $Y$, with all vertices except for $u$ contained in $Y' = V(G) \setminus (X' \cup B)$.

Consider a graph $G''$ defined as follows: we take the graph $G'[V(G') \setminus X']$ and add a terminal $s$ and arcs $(s, w)$ for all $w \in N^+_{G'}(X') = (B \setminus \{v\}) \cup (N^+_G(v) \setminus X')$. Let $B'$ be a minimum size $s - Y$ separator in $G''$.

We claim that $B'$ is an $X - Y$ separator in $G$. Let $P$ be an arbitrary $XY$-path in $G$ and let $u$ be the last (closest to $Y$) vertex of $B$ on $P$. Then in $G''$ there exists a shortened version $P'$ of $P$: if $u = v$ and $w$ is the vertex directly after $u$ on $P$, then $P'$ starts with the arc $(s, w)$ and then follows $P$ to $Y$ (observe that $w \notin X'$, as $v$ was the last vertex of $B$ on $P$), while if $u \neq v$, then $P'$ starts with the arc $(s, u)$ and then follows $P$ to $Y$. As $B'$ has to intersect $P'$, then $B'$ also intersects $P$.

Moreover, we claim that $B'$ is behind $B$. This follows directly from the fact that $G''$ does not contain $X'$, so $X'$ is still reachable from $X$ in $G \setminus B'$.

As $B$ is an important separator, $B'$ is behind $B$, and $B' \neq B$ (as $v \in B \setminus B'$), we have that $|B'| > |B|$. Therefore, there exists a family $\mathcal{P}$ of at least $|B|+1$ $sY$-paths in $G''$ that have pairwise disjoint interiors. Observe that all these paths are disjoint with $X'$ by the construction of $G''$. Each path from $\mathcal{P}$ that starts with an arc $(s,w)$ for $w \in N^+_G(v) \setminus X'$ is present (with the first vertex replaced by an arbitrary vertex of $X$) in $G'$ as well. Moreover, each other path starts with an arc $(s,u)$ for $u \in B$; in $G'$ such a path can be concatenated with the $Xu$-path $P^X_u$. All paths $P^X_u$ are entirely contained in $X'$ except for the endpoint $u$, so we obtain the desired family of $XY$-paths in $G'$. □

We use the technique of shadows [13, 6, 5] to identify vertices separated from all sources in a given MULTICUT IN DAGS instance. We note that we do not use the full power of the shadow removal technique in directed graphs: the delicate part of the results of Chitnis, Hajiaghayi, and Marx [6, 5] is to remove forward and reverse shadows at once; in our work we need to remove only one type of shadows, namely, forward ones.

We now recall the necessary definitions of the shadow removal technique from [6] and the improved bounds of [5]. Although we give full definitions for completeness, we will chiefly need Definition 2.10 and Lemma 2.11 in the rest of the paper.

DEFINITION 2.7 (shadow [6, Definition 2.3]). *Let $G$ be a directed graph and $T \subseteq V(G)$ be a set of terminals. Let $Z \subseteq V(G)$ be a subset of vertices. Then for $v \in V(G)$ we say that*

1. *$v$ is in the* forward shadow *of $Z$ (with respect to $T$) if $Z$ is a $T - v$ separator in $G$, and*
2. *$v$ is in the* reverse shadow *of $Z$ (with respect to $T$) if $Z$ is a $v - T$ separator in $G$.*

DEFINITION 2.8 (thin [6, Definition 4.4]). *Let $G$ be a directed graph and $T \subseteq V(G)$ a set of terminals. We say that a set $Z \subseteq V(G)$ is* thin *in $G$ if there is no $v \in Z$ such that $v$ belongs to the reverse shadow of $Z \setminus \{v\}$ with respect to $T$.*

THEOREM 2.9 (derandomized random sampling [5][1]). *There is an algorithm that, given a directed graph $G$, a set of terminals $T \subseteq V(G)$, and an integer $p$, produces in time $O^*(2^{O(p^2)})$ a family $\mathcal{A}$ of size $2^{O(p^2)} \log |V(G)|$ of subsets of $V(G) \setminus T$ such that the following holds. Let $Z \subseteq V(G) \setminus T$ be a thin set with $|Z| \leq p$ and let $Y$ be a set such that for every $v \in Y$ there is an important $v - T$ separator $Z_v \subseteq Z$. For every such pair $(Z, Y)$ there exists a set $A \in \mathcal{A}$ such that $A \cap Z = \emptyset$ but $Y \subseteq A$.*

We use Theorem 2.9 to identify vertices separated from all sources in a given MULTICUT IN DAGS instance.

DEFINITION 2.10 (source shadow). *Let $(G, \mathcal{T}, p)$ be a* MULTICUT IN DAGS *instance and $Z \subseteq V(G)$ be a set of nonterminals in $G$. We say that $v \in V(G)$ is in* source shadow *of $Z$ if $Z$ is a $T^s - v$ separator.*

LEMMA 2.11 (derandomized random sampling for source shadows). *There is an algorithm that, given a* MULTICUT IN DAGS *instance $(G, \mathcal{T}, p)$, produces in time $O^*(2^{O(p^2)})$ a family $\mathcal{A}$ of size $2^{O(p^2)} \log |V(G)|$ of subsets of nonterminals of $G$ such that if $(G, \mathcal{T}, p)$ is a YES instance and $Z$ is the lex-min solution to $(G, \mathcal{T}, p)$, then there exists $A \in \mathcal{A}$ such that $A \cap Z = \emptyset$ and all vertices of source shadows of $Z$ in $G$ are contained in $A$.*

[1] We cite here improved yet unpublished bounds of [5]. See [6, Theorem 4.1 and section 4.3] for the same result but with double-exponential bounds.

*Proof.* Let $Y$ be the set of vertices in source shadow of $Z$. To prove the lemma it is sufficient to apply Theorem 2.9 for the graph $G^{\mathrm{rev}}$ with terminals $T^s$ and budget $p$. Thus, we need to prove that the pair $(Z, Y)$ satisfies properties given in Theorem 2.9.

First, assume that $Z$ is not thin in $G^{\mathrm{rev}}$ w.r.t. $T^s$. Let $v \in Z$ be a witness: $Z \setminus \{v\}$ is a $v - T^s$ separator in $G^{\mathrm{rev}}$. We infer that $S(G \setminus (Z \setminus \{v\}), v) = \emptyset$ and $Z \setminus \{v\}$ is a multicut in $(G, \mathcal{T})$, a contradiction to the choice of $Z$.

Second, take an arbitrary vertex $u \in Y$, that is, $u$ is in source shadow of $Z$ in $G$. Let $B \subseteq Z$ be the set of those vertices $v \in Z$ for which there exists a $vu$-path in $G \setminus (Z \setminus \{v\})$. Clearly, $B$ is a $u - T^s$ separator in $G^{\mathrm{rev}}$. We claim that it is an important one.

If $B$ is not a minimal $u - T^s$ separator in $G^{\mathrm{rev}}$, let $v \in B$ be such that $B \setminus \{v\}$ is a $u - T^s$ separator in $G^{\mathrm{rev}}$; then $B \setminus \{v\}$ is a $v - T^s$ separator in $G^{\mathrm{rev}}$ (as there is a $vu$-path in $G \setminus (Z \setminus \{v\})$) and $Z \setminus \{v\}$ is a multicut in $(G, \mathcal{T})$, a contradiction to the choice of $Z$.

Assume then that there exists a $u - T^s$ separator $B'$ in $G^{\mathrm{rev}}$ that is *behind* $B$ and $|B'| \leq |B|$, $B' \neq B$. We claim that $Z' = (Z \setminus B) \cup B'$ is a multicut in $(G, \mathcal{T})$. This would lead to a contradiction with the choice of $Z$, as $|Z'| \leq |Z|$ and $Z'$ is smaller in the lexicographical order than $Z$. Assume then that $Z'$ is not a multicut in $(G, \mathcal{T})$, that is, there is an $s_i t_i$-path $P$ in $G \setminus Z'$ for some $1 \leq i \leq r$. As $Z$ is a multicut in $(G, \mathcal{T})$, $P$ contains at least one vertex $v \in B \setminus B' = Z \setminus Z'$. By the choice of $B$ and the fact that $B'$ is behind $B$ in $G^{\mathrm{rev}}$, we infer that there exists a $vu$-path in $G \setminus Z'$. As $P$ contains $v$, there exists an $s_i u$-path in $G \setminus Z'$, a contradiction to fact that $B'$ is an $u - T^s$ separator in $G^{\mathrm{rev}}$. This concludes the proof of the lemma. □

**3. The algorithm.** We now present the FPT algorithm for Multicut in DAGs. The algorithm is an intricate, multistep branching process using several tools, hence we split the presentation into several parts. In section 3.1, we give the basic notions and results and introduce the potential function we will use to analyze the branching process. In section 3.2, we present an important *degree reduction* process, by which we can arrange so that the source terminals have bounded degree. Section 3.3 contains an overview of the branching process, and finally section 3.4 gives the full details and correctness proofs. A diagram of the algorithm is given in Figure 1.

**3.1. Potential function and simple operations.** Our algorithm consists of a number of branching steps. To measure the progress of the algorithm, we introduce the following potential function.

Definition 3.1 (potential). *Given a* Multicut in DAGs *instance* $\mathcal{I} = (G, \mathcal{T}, p)$, *we define its potential* $\phi(\mathcal{I})$ *as* $\phi(\mathcal{I}) = (r+1)p - \sum_{i=1}^{r} cut_G(s_i, t_i)$.

Observe that if $\mathcal{I} = (G, \mathcal{T}, p)$ is a Multicut in DAGs instance, in which $\mathrm{cut}(s_i, t_i) > p$ for some $(s_i, t_i) \in \mathcal{T}$, then we can immediately conclude that $\mathcal{I}$ is a NO instance. Therefore, w.l.o.g. we can henceforth assume that $\mathrm{cut}(s_i, t_i) \leq p$ for all $(s_i, t_i) \in \mathcal{T}$ in all the appearing instances of Multicut in DAGs.

In many places we perform the following simple operations on Multicut in DAGs instances $(G, \mathcal{T}, p)$. We formalize their properties in subsequent lemmata.

Definition 3.2 (killing a vertex). *For a* Multicut in DAGs *instance* $(G, \mathcal{T}, p)$ *and a nonterminal vertex* $v$ *of* $G$, *by* killing the vertex $v$ *we mean the following operation: we delete the vertex* $v$ *and decrease* $p$ *by one.*

Definition 3.3 (bypassing a vertex). *For a* Multicut in DAGs *instance* $(G, \mathcal{T}, p)$ *and a nonterminal vertex* $v$ *of* $G$, *by* bypassing the vertex $v$ *we mean the following operation: we delete the vertex* $v$ *and for each in-neighbor* $v^-$ *of* $v$ *and each out-neighbor* $v^+$ *of* $v$ *we add an arc* $(v^-, v^+)$.

LEMMA 3.4. *Let $\mathcal{I}' = (G', \mathcal{T}, p-1)$ be obtained from* MULTICUT IN DAGs *instance $\mathcal{I} = (G, \mathcal{T}, p)$ by killing a vertex $v$. Then $\mathcal{I}'$ is a YES instance if and only if $\mathcal{I}$ is a YES instance that admits a solution that contains $v$. Moreover, $\phi(\mathcal{I}') < \phi(\mathcal{I})$.*

*Proof.* Let $Z$ be a multicut in $\mathcal{I}$ that contains $v$. As $G \setminus Z = G' \setminus (Z \setminus \{v\})$, $Z \setminus \{v\}$ is a multicut in $\mathcal{I}'$ of size $|Z|-1$. In the other direction, if $Z$ is a multicut in $\mathcal{I}'$, then $G' \setminus Z = G \setminus (Z \cup \{v\})$ and $Z \cup \{v\}$ is a multicut in $\mathcal{I}$ of size $|Z|+1$. To see that the potential strictly decreases, note that $\mathrm{cut}_{G'}(s_i, t_i) \geq \mathrm{cut}_G(s_i, t_i) - 1$ for all $1 \leq i \leq r$. □

LEMMA 3.5. *Let $\mathcal{I}' = (G', \mathcal{T}, p)$ be obtained from* MULTICUT IN DAGs *instance $\mathcal{I} = (G, \mathcal{T}, p)$ by bypassing a vertex $v$. Then*

1. *any multicut in $\mathcal{I}'$ is a multicut in $\mathcal{I}$ as well;*
2. *any multicut in $\mathcal{I}$ that does not contain $v$ is a multicut in $\mathcal{I}'$ as well;*
3. *$S(G, u) = S(G', u)$ for any $u \in V(G') = V(G) \setminus \{v\}$;*
4. *$\phi(\mathcal{I}') \leq \phi(\mathcal{I})$.*

*Proof.* The lemma follows from the following observations on relations between paths in $G$ and $G'$. For a path $P$ in $G$ whose first and last points are different from $v$, we define $P_{G'}$ as $P$ with a possible occurrence of $v$ removed. By the definition of $G'$, $P_{G'}$ is a path in $G'$. In the other direction, for a path $P$ in $G'$, we define $P_G$ as a path obtained from $P$ by inserting the vertex $v$ between any consecutive vertices $v_i, v_{i+1}$ for which $(v_i, v_{i+1}) \in E(G') \setminus E(G)$. Since $G$ and $G'$ are acyclic, the vertex $v$ is inserted at most once. By the construction of $G'$, $P_G$ is a path in $G$.

Now, for a multicut $Z$ in $\mathcal{I}'$ and an arbitrary $s_i t_i$-path $P$ in $G$, the path $P_{G'}$ is intersected by $Z$; thus $P$ is intersected by $Z$ as well and $Z$ is a multicut in $\mathcal{I}$. For a multicut $Z$ in $\mathcal{I}$ with $v \notin Z$, and an arbitrary $s_i t_i$-path $P$ in $G'$, the path $P_G$ is intersected by $Z$. As $v \notin Z$, we infer that $P$ is intersected by $Z$ as well and $Z$ is a multicut in $\mathcal{I}'$.

To prove the third claim, note that for any $u \in V(G')$, any $s_i u$-path $P$ in $G$ yields an $s_i u$-path $P_{G'}$ in $G'$ and vice versa. Finally, the last claim follows from the fact that any family $\mathcal{P}$ of $s_i t_i$-paths in $G$ with pairwise disjoint sets of internal vertices can be transformed into a similar family $\mathcal{P}' = \{P_{G'} : P \in \mathcal{P}\}$ in $G'$. Thus $\mathrm{cut}_{G'}(s_i, t_i) \geq \mathrm{cut}_G(s_i, t_i)$ for $1 \leq i \leq r$. □

We note that bypassing a vertex corresponds to the torso operation of Chitnis, Hajiaghayi, and Marx [6] and, if we perform a series of bypass operations, the result does not depend on their order.

LEMMA 3.6. *Let $\mathcal{I} = (G, \mathcal{T}, p)$ and $X \subseteq V(G)$ be a subset of nonterminals of $G$. Let $\mathcal{I}' = (G', \mathcal{T}, p)$ be obtained from $\mathcal{I}$ by bypassing all vertices of $X$ in an arbitrary order. Then $(u, v) \in E(G')$ for $u, v \in V(G') = V(G) \setminus X$ if and only if there exists a $uv$-path in $G$ with internal vertices from $X$ (possibly consisting only of an arc $(u, v)$). In particular, $\mathcal{I}'$ does not depend on the order in which the vertices of $X$ are bypassed.*

*Proof.* We perform induction with respect to the size of the set $X$. For $X = \emptyset$ the lemma is trivial.

Let $\mathcal{I}'' = (G'', \mathcal{T}, p)$ be an instance obtained by bypassing all vertices of $X \setminus \{w\}$ in $\mathcal{I}$ in an arbitrary order for some $w \in X$. Take $u, v \in V(G') = V(G) \setminus X$. Assume first that $(u, v) \in E(G')$. By the definition of bypassing, $(u, v) \in E(G'')$ or $(u, w), (w, v) \in E(G'')$. In the first case there exists a $uv$-path in $G$ with internal vertices from $X \setminus \{w\}$ by the induction hypothesis. In the second case, by the induction hypothesis, there exist a $uw$-path and a $wv$-path in $G$, both with internal vertices from $X \setminus \{w\}$; their concatenation is the desired $uv$-path (recall that $G$ is acyclic).

In the other direction, let $P$ be a $uv$-path in $G$ with internal vertices in $X$. If $w$ does not lie on $P$, by the induction hypothesis $(u, v) \in E(G'')$. Otherwise, $P$ splits

into a $uw$-path and a $wv$-path, both with internal vertices in $X \setminus \{w\}$. By the induction hypothesis $(u,w),(w,v) \in E(G'')$. By the definition of bypassing, $(u,v) \in E(G')$ and the lemma is proved. □

**3.2. Degree reduction.** We now introduce the second main tool used in the algorithm (the first one being the source shadow reduction of Lemma 2.11). In an instance $(G,\mathcal{T},p)$, let $B_i$ be the $s_i - t_i$ mincut closest to $s_i$ and let $Z$ be a solution. If we know that a $vt_i$-path survives in $G \setminus Z$ for some $v \in B_i$, we may add an arc $(v,t_i)$ and then bypass the vertex $v$, strictly increasing the value $\mathrm{cut}_G(s_i,t_i)$ (and thus decreasing the potential) by Lemma 2.6. Therefore, we can branch: we either guess the pair $(i,v)$ or guess that none such exist; in the latter branch we do not decrease potential but instead we may modify the set of arcs incident to the sources to get some structure, as formalized in the following definition.

DEFINITION 3.7 (degree-reduced graph). *For a* MULTICUT IN DAGS *instance* $(G,\mathcal{T},p)$ *the* degree-reduced *graph* $G^*$ *is a graph constructed as follows. For* $1 \le i \le r$, *let* $B_i$ *be the* $s_i - t_i$ *mincut closest to* $s_i$. *We start with* $V(G^*) = V(G)$, $E(G^*) = E(G \setminus T^s)$ *and then, for each* $1 \le i \le r$, *we add an arc* $(s_i,v)$ *for all* $v \in B_i$ *and for all* $v \in \bigcup_{1 \le i' \le r} B_{i'}$ *for which* $s_i \in S(G,v)$ *but* $v$ *is not reachable from* $B_i$ *in* $G$.

The following two lemmata formalize the properties of the degree-reduced graph and the aforementioned branching step. Recall that we assume that each vertex $s_i$ ($t_i$) has in- (out-) degree zero.

LEMMA 3.8 (properties of the degree-reduced graph). *For any* MULTICUT IN DAGS *instance* $\mathcal{I} = (G,\mathcal{T},p)$ *and the degree-reduced graph* $G^*$ *of* $\mathcal{I}$, *the following holds:*

1. $|N^+_{G^*}(T^s)| \le rp$.
2. *For each* $1 \le i \le r$, $B_i$ *is the* $s_i - t_i$ *mincut closest to* $s_i$ *in* $G^*$.
3. $\phi(\mathcal{I}') = \phi(\mathcal{I})$, *where* $\mathcal{I}' = (G^*,\mathcal{T},p)$.
4. $Z \subseteq V(G)$ *is a multicut in* $(G^*,\mathcal{T})$ *if and only if* $Z$ *is a multicut in* $(G,\mathcal{T})$ *satisfying the following property: for each* $1 \le i \le r$, *for each* $v \in B_i$, *the vertex* $v$ *is either in* $Z$ *or* $Z$ *is an* $v - t_i$ *separator; in particular,* $\mathcal{I}'$ *is a YES instance if and only if* $\mathcal{I}$ *is a YES instance that admits a solution satisfying the above property.*
5. *For each* $v \in V(G)$ *we have* $S(G^*,v) \subseteq S(G,v)$; *moreover, if* $(s_i,v)$ *is an arc in* $G^*$ *for some* $1 \le i \le r$, *then* $S(G^*,v) = S(G,v)$.

*Proof.* For claim 1, note that for each $1 \le i \le r$, $N^+_{G^*}(s_i) \subseteq \bigcup_{1 \le i' \le r} B_{i'}$ and $|B_{i'}| = \mathrm{cut}_G(s_{i'},t_{i'}) \le p$.

For Claim 2, we first prove that $B_i$ is an $s_i - t_i$ separator in $G^*$. Assume otherwise; let $P$ be an $s_it_i$-path in $G^*$ that avoids $B_i$. Let $(s_i,v)$ be the first arc on $P$. By the construction of $G^*$, in $G$ the vertex $v$ is reachable from $s_i$ but not from $B_i$. Therefore, there exists an $s_iv$-path in $G \setminus B_i$; together with $P$ truncated by $s_i$ it gives an $s_it_i$-path in $G \setminus B_i$, a contradiction.

Now note that for each $1 \le i \le r$, there exist $\mathrm{cut}_G(s_i,t_i)$ $s_it_i$-paths in $G$ with pairwise disjoint interiors; each path visits a different vertex of $B_i$. These paths can be shortened in $G^*$ using arcs $(s_i,v)$ for $v \in B_i$; thus, $B_i$ is an $s_i - t_i$ separator in $G^*$ of minimum size. The fact that it is an important $t_i - s_i$ separator in $G^{\mathrm{rev}}$ follows from the fact that $B_i \subseteq N^+_{G^*}(s_i)$.

Claim 3 follows directly from claim 2, as $\mathrm{cut}_G(s_i,t_i) = \mathrm{cut}_{G^*}(s_i,t_i)$ for all $1 \le i \le r$.

For claim 4, let $Z$ be a multicut in $(G^*,\mathcal{T})$. Take arbitrary $1 \le i \le r$ and let $P$ be an arbitrary $s_it_i$-path in $G$. This path intersects $B_i$; let $v$ be the last (closest to

$t_i$) vertex of $B_i$ on $P$. Let $P^*$ be an $s_it_i$-path in $G^*$ defined as follows: we start with the arc $(s_i, v)$ and then we follow $P$ from $v$ to $t_i$. As $Z$ is a multicut in $(G^*, \mathcal{T})$, $Z$ intersects $P^*$. We infer that $Z$ intersects $P$ and $Z$ is a multicut in $G$. Moreover, for each $1 \leq i \leq B_i$ and $v \in B_i$, if $v \notin Z$, then $Z$ is a $v - t_i$ separator in $G^*$ (and in $G$ as well, as $G$ and $G^*$ differ only on arcs incident to the sources), as otherwise $Z$ would not intersect an $s_it_i$-path in $G^*$ that starts with the arc $(s_i, v)$.

In the second direction, let $Z$ be a multicut in $(G, \mathcal{T})$ that satisfies the conditions given in claim 4. Let $P^*$ be an arbitrary $s_it_i$-path in $G^*$. As $B_i$ is an $s_i - t_i$ separator in $G^*$, $P^*$ intersects $B_i$; let $v$ be the last (closest to $t_i$) vertex of $B_i$ on $P^*$. Note that the part of the path $P^*$ from $v$ to $t_i$ (denote it by $P_v$) is present also in the graph $G$. By the properties of $Z$, $v \in Z$ or $Z$ intersects $P_v$. Thus $Z$ intersects $P^*$ and $Z$ is a multicut in $(G^*, \mathcal{T})$.

To see the first part of claim 5 note that if $(s_i, v)$ is an arc in $G^*$, then $s_i \in S(G, v)$: clearly this is true for $v \in B_i$, and otherwise $s_i \in S(G, v)$ is one of the conditions required to add arc $(s_i, v)$. The second part follows directly from the construction: if $(s_i, v)$ is an arc in $G^*$, then $v \in B_{i'}$ for some $1 \leq i' \leq r$. If $s_{i''} \in S(G, v)$, then either $v$ is reachable from some vertex of $B_{i''}$ in $G^*$ or the arc $(s_{i''}, v)$ is present in $G^*$. □

We note that in the definition of the degree-reduced graph, the arcs between a source $s_i$ and vertices in $B_{i'}$ for $i \neq i'$ are added only to ensure claim 5. For the remaining claims, as well as the branching described at the beginning of the section (formalized in the subsequent lemma) it would suffice to add only arcs $(s_i, v)$ for $1 \leq i \leq r$ and $v \in B_i$.

LEMMA 3.9. *There exists an algorithm that, given a* MULTICUT IN DAGS *instance $\mathcal{I} = (G, \mathcal{T}, p)$, in polynomial time generates a sequence of instances $(\mathcal{I}_j = (G_j, \mathcal{T}_j, p_j))_{j=1}^{d}$ satisfying the following properties. Let $\mathcal{I}_0 = (G^*, \mathcal{T}, p)$;*

1. *if $Z$ is a multicut in $\mathcal{I}_j$ for some $0 \leq j \leq d$, then $Z \subseteq V(G)$ and $Z$ is a multicut in $\mathcal{I}$ too;*
2. *for any multicut $Z$ in $\mathcal{I}$, there exists $0 \leq j \leq d$ such that $Z$ is a multicut in $\mathcal{I}_j$ too;*
3. *for each $1 \leq j \leq d$, $p_j = p$, $\mathcal{T}_j = \mathcal{T}$ and $\phi(\mathcal{I}_j) < \phi(\mathcal{I})$;*
4. *$d \leq rp$.*

*Proof.* Let $B_i$ be as in Definition 3.7. Informally speaking, we guess an index $1 \leq i \leq r$ and a vertex $v \in B_i$ such that $t_i$ is reachable from $v$ in $G \setminus Z$, where $Z$ is a solution to $\mathcal{I}$ (in particular, $v \notin Z$). If we have such $v$, we can add an arc $(v, t_i)$ and then bypass $t_i$; by the choice of $B_i$ and Lemma 2.6, the value $\mathrm{cut}(s_i, t_i)$ strictly increases during this operation. The last branch—where such a choice $(i, v)$ does not exists—corresponds to the degree-reduced graph $G^*$. We now proceed to the formal arguments.

For $1 \leq i \leq r$ and $v \in B_i$ we define the graph $G_{i,v}$ as follows: we first add an arc $(v, t_i)$ to $G$ and then bypass the vertex $v$. We now apply Lemma 2.6 for $t_i - s_i$ cuts in the graph $G^{\mathrm{rev}}$: $B_i$ is the unique minimum size important $t_i - s_i$ separator, $v \in B_i$, and $G^{\mathrm{rev}}_{i,v}$ contains the same set of vertices and a superset of arcs of the graph $G'$ from the statement of the lemma. Therefore $\mathrm{cut}_{G_{i,v}}(s_i, t_i) > \mathrm{cut}_G(s_i, t_i)$. Moreover, $\mathrm{cut}_{G_{i,v}}(s_{i'}, t_{i'}) \geq \mathrm{cut}_G(s_{i'}, t_{i'})$ for $i' \neq i$, as adding an arc and bypassing a vertex cannot decrease the size of the minimum separator. Therefore $\phi((G_{i,v}, \mathcal{T}, p)) < \phi(\mathcal{I})$; we output $(G_{i,v}, \mathcal{T}, p)$ as one of the output instances $\mathcal{I}_j$. Clearly, $d = \sum_{i=1}^{r} |B_i| \leq rp$. To finish the proof of the lemma we need to show the equivalence stated in the first two points of the statement.

In one direction, note that as the graphs $G_{i,v}$ are constructed from $G$ by adding an arc and bypassing a vertex, any multicut in $(G_{i,v}, \mathcal{T})$ is a multicut in $(G, \mathcal{T})$ as well. Moreover, by Lemma 3.8, claim 4, any multicut of $\mathcal{I}_0$ is a multicut of $\mathcal{I}$ as well.

In the other direction, let $Z$ be a solution to $\mathcal{I}$. Consider two cases. First assume that there exists $1 \leq i \leq r$ and $v \in B_i$ such that $v \notin Z$ and $Z$ is *not* a $v - t_i$ separator. As $Z$ is a multicut in $(G, \mathcal{T})$, $Z$ is a $s_i - v$ separator in $G$. Therefore $Z$ is also a multicut in a graph $G$ with the arc $(v, t_i)$ added. As $v \notin Z$, by Lemma 3.5, $Z$ is a multicut in $(G_{i,v}, \mathcal{T})$. In the second case, if for each $1 \leq i \leq r$ and $v \in B_i$, we have $v \in Z$ or $Z$ is a $v - t_i$ separator in $G$, we conclude that $Z$ is a multicut in $(G^*, \mathcal{T})$ by Lemma 3.8, claim 4. □

**3.3. Overview on the branching step.** In order to prove Theorem 1.1, we show the following lemma that encapsulates a single branching step of the algorithm.

LEMMA 3.10. *There exists an algorithm that, given a* MULTICUT IN DAGS *instance $\mathcal{I} = (G, \mathcal{T}, p)$ with $|\mathcal{T}| = r$, in time $O^*(2^{r+O(p^2)})$ either correctly concludes that $\mathcal{I}$ is a NO instance or computes a sequence of instances $(\mathcal{I}_j = (G_j, \mathcal{T}_j, p_j))_{j=1}^{d}$ such that*

1. *$\mathcal{I}$ is a YES instance if and only if at least one instance $\mathcal{I}_j$ is a YES instance;*
2. *for each $1 \leq j \leq d$, $V(G_j) \subseteq V(G)$, $p_j \leq p$, $\mathcal{T}_j = \mathcal{T}$, and $\phi(\mathcal{I}_j) < \phi(\mathcal{I})$;*
3. *$d \leq 2^{r+O(p^2)} rp \log |V(G)|$.*

The algorithm of Theorem 1.1 applies Lemma 3.10 and solves the output instances recursively.

*Proof of Theorem* 1.1. Let $\mathcal{I} = (G, \mathcal{T}, p)$ be a MULTICUT IN DAGS instance. Clearly, if $\phi(\mathcal{I}) < 0$, then $\mathrm{cut}(s_i, t_i) > p$ for some $1 \leq i \leq r$ and the instance is a NO instance. Otherwise, we apply Lemma 3.10 and solve the output instances recursively. Note that the potential of $\mathcal{I}$ is an integer bounded by $(r+1)p$; thus the search tree of the algorithm has depth at most $(r+1)p$. Using a simple fact that for $k, n > 1$ we have

$$\log^k n = 2^{k \log\log n} \leq 2^{\frac{2}{3}k^{3/2} + \frac{1}{3}(\log\log n)^3} = 2^{\frac{2}{3}k^{3/2}} n^{o(1)},$$

we obtain that the number of leaves of the search tree is bounded by

$$\begin{aligned}\left(2^{r+O(p^2)} rp \log |V(G)|\right)^{(r+1)p} &= 2^{O(r^2 p)} \cdot 2^{O(rp^3)} \cdot 2^{O(r^{3/2}p^{3/2})} |V(G)|^{o(1)} \\ &= O^*(2^{O(rp(r+p^2))}).\end{aligned}$$

The last equality follows from the fact that $r^{3/2}p^{3/2} \leq r^2 p + rp^3$. □

In rough overview of the proof of Lemma 3.10, we describe a sequence of steps, as illustrated in Figure 1, where in each step, either the potential of the instance is decreased or more structure is forced onto the instance. For example, consider Lemma 3.9. The result is a branching into polynomially many branches, where in every branch but one the potential strictly decreases, and in the remaining branch, the degrees of the source terminals are bounded. Thus we may treat this step as "creating" a degree-reduced instance. The ultimate goal is to pin down a set of not too many vertices, at least one of which is guaranteed to be deleted in an optimal solution. In total, the entire process can be seen as producing some $f(p, r)$ branches with new instances, with a strictly decreased potential in each branch.

In somewhat more detail, let $Z$ be the lex-min solution to $\mathcal{I}$. To illustrate the principles, consider first a single terminal $s \in T^s$, and assume that we know (or have guessed) that there is a vertex $v \in Z$ such that $S(G, v) = \{s\}$. Let us suppose that $V(G, \{s\})$ is too large to branch on directly; we reduce its size by a sequence of modifications. We first observe that for every $v \in V(G, \{s\})$ with $v \notin Z$, either $s$

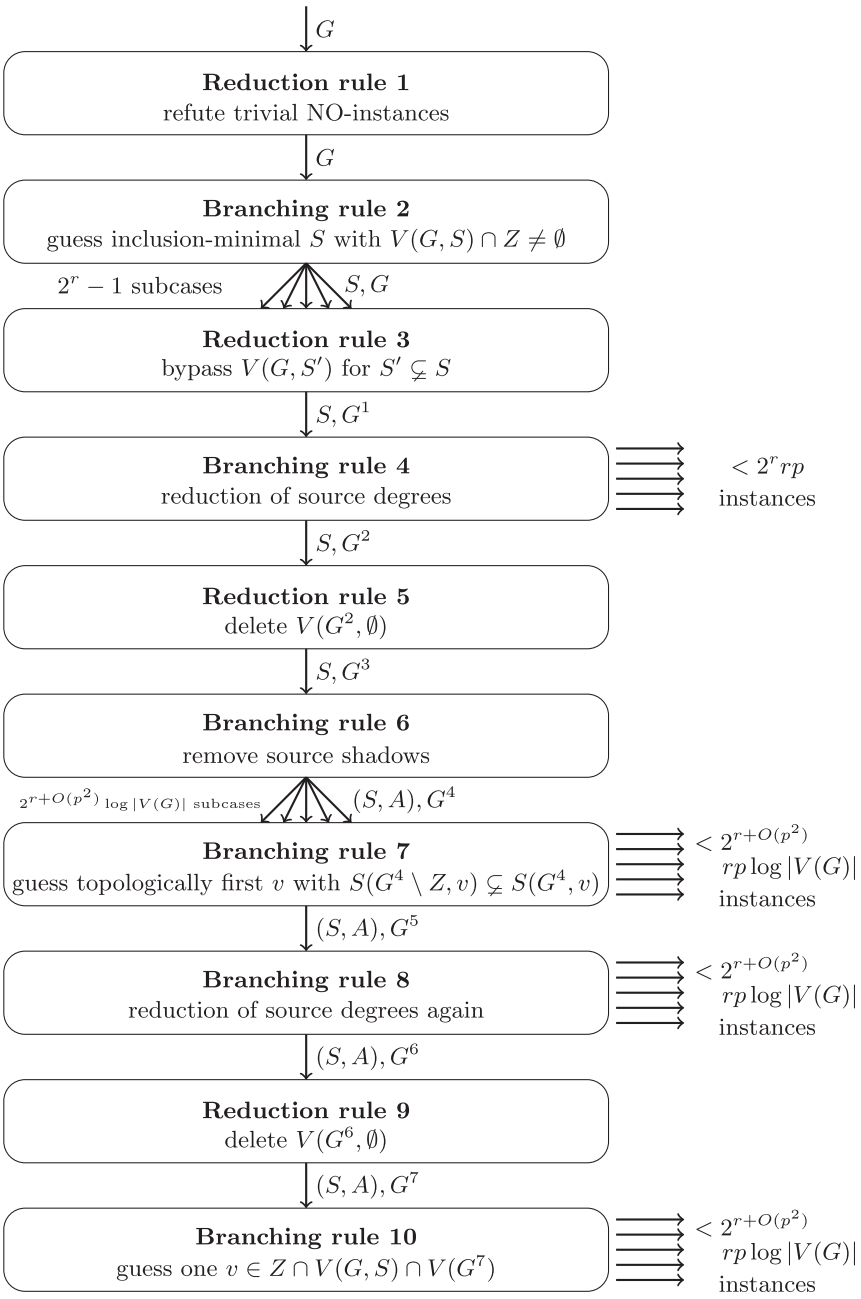


FIG. 1. *A summary of one branching step.*

reaches $v$ in $G \setminus Z$ or $v$ is in source shadow of $Z$. Lemma 2.11 lets us guess a set $A$ which contains all vertices of the latter type; we apply this lemma, bypass the vertices $A$, and assume in what follows that every $v \in V(G, \{s\})$ is either contained in $Z$ or reached by $s$ in $G \setminus Z$.

The next observation is that under this assumption, we may (1) add an arc $sv$ for every $v \in V(G, \{s\})$, and (2) remove every arc $uv$ where $u, v \in V(G, \{s\})$. It can be verified that this preserves $Z$ as a (lex-min) solution, without creating any new

solutions. Furthermore we get that for every $v \in V(G, \{s\})$, the unique in-neighbor of $v$ is $s$. Now we apply the degree reduction of section 3.2. This leads to a graph $G_0$ where at most $p$ vertices are out-neighbors of $s$; the remaining vertices from $V(G, \{s\})$ now belong to $V(G_0, \emptyset)$ and can be discarded. We may now as a final branching select one remaining out-neighbor of $s$ and kill it, yielding at most $p$ further branches with strictly decreased potential. (This sketch corresponds roughly to steps 2, 6, and 8–10 of the branching process.)

For the general case, assume that we have guessed a set $S \subseteq T^s$ such that there is some $v \in Z$ with $S(G, v) = S$, but no $v' \in Z$ with $S(G, v') \subsetneq S$; we may bypass any vertex $u$ with $S(G, u) \subsetneq S$. By appropriately combining degree reduction with shadow removal, we may further assume that no vertex in $V(G, S)$ is in source-shadow of $Z$ and that the sources $S$ have bounded degree. We wish to follow the scheme outlined above for a single terminal $s$. However, it is not yet safe to add an arc $sv$ for *every* $s \in S$ and $v \in V(G, S)$, as there may be some vertices $v \in V(G, S)$ such that $S(G \setminus Z, v) = S'$, where $S' \neq \emptyset$, $S' \subsetneq S$. Let us refer to such a vertex $v'$ where furthermore $v' \notin Z$, as being *modified* by $Z$.

We now reach a key point of the paper (Lemma 3.12 and Branching Rule 7). Let $v$ be the *earliest* vertex of $V(G, S)$ which is modified by $Z$, as measured by $\preceq$; assume that such a vertex exists. We make a few observations. First, if $u$ is a non-terminal in-neighbor of $v$, then $u \in Z$, as otherwise $S(G \setminus Z, u) = S$ by assumption, contradicting that $S(G \setminus Z, v) \subsetneq S$. Second, $v$ must have at least one terminal in-neighbor, as otherwise $S(G \setminus Z, v) = \emptyset$, contradicting that $v$ is not in source shadow. Finally, not every $s \in S$ is an in-neighbor of $v$, hence $v$ must have at least one non-terminal in-neighbor $u$ in $G$, as $S(G, v) = S$. Hence, if any modified vertex exists in $V(G, S)$, then the earliest such vertex is contained in $N^+(S)$, which has bounded size by degree reduction, and furthermore the identification of this vertex $v$ allows us to kill at least one vertex $u \in N^-(v)$, thereby strictly decreasing the potential of the instance. This finishes the case where $V(G, S)$ contains some modified vertex.

Now, if $V(G, S)$ contains no modified vertices, then we can indeed follow the scheme set out for the single-terminal case: Add all arcs $sv$ for $s \in S$, $v \in V(G, S)$; remove all arcs within $V(G, S)$; use degree reduction to move the majority of $V(G, S)$ into $V(G_0, \emptyset)$; and branch over the killing of one vertex from the set of at most $rp$ remaining vertices. This finishes the branching process.

**3.4. Branchings and reductions.** We now proceed with the formal proof of Lemma 3.10. The proof contains a sequence of *branching rules* (when we generate a number of subcases, some of them already ready to output as one instance $\mathcal{I}_j$) or *reduction rules* (when we reduce the graph without changing the answer). To make the algorithm easier to follow, we embed all branching and reduction rules in appriopriately numbered environments. Moreover, a diagram of the branching step is given in Figure 1.

If the input instance $\mathcal{I}$ is YES instance, by $Z$ we denote its lex-min solution. Whenever we perform a branching or reduction step, in the new instance we consider the topological order that is induced by the old one; all the reductions and branchings add arcs only directed from vertices smaller in $\preceq$ to bigger. This also ensures that during the course of the algorithm all the directed graphs in the instances are acyclic.

We start with the obvious rule that was already mentioned in section 3. Then, we roughly localize one vertex of $Z$.

REDUCTION RULE 1. *If $cut_G(s_i, t_i) > p$ (in particular, if $(s_i, t_i) \in E(G)$) for some $1 \leq i \leq r$, conclude that $\mathcal{I}$ is a NO instance.*

BRANCHING RULE 2. *Branch into $2^r - 1$ subcases, labeled by nonempty sets $S \subseteq T^s$. In the case labeled $S$ we assume that $Z$ contains a vertex $v$ with $S(G, v) = S$, but no vertex $v'$ with $S(G, v')$ being a proper subset of $S$.*

As $Z$ is a lex-min solution (in case of $\mathcal{I}$ being a YES instance), $Z$ cannot contain any vertex $v$ with $S(G, v) = \emptyset$. In each branch we can bypass some vertices.

REDUCTION RULE 3. *In each subcase, as long as there exists a nonterminal vertex $u \in V(G)$ with $S(G, u) \subsetneq S$ bypass $u$. Let $(G^1, \mathcal{T}, p)$ be the reduced instance.*

By Lemma 3.5, an application of the above rule cannot turn a NO instance into a YES instance. Moreover, in the branch where $S$ is guessed correctly, $Z$ remains the lex-min solution to $(G^1, \mathcal{T}, p)$. By Lemma 3.5, $\phi((G^1, \mathcal{T}, p)) \leq \phi(\mathcal{I})$.

We now apply the reduction of source degrees.

BRANCHING RULE 4. *In each subcase, let $S$ be its label and $(G^1, \mathcal{T}, p)$ be the instance. Invoke Lemma 3.9 on the instance $(G^1, \mathcal{T}, p)$. Output all instances $\mathcal{I}_j$ for $1 \leq j \leq d$ as part of the output instances in Lemma 3.10. Keep the instance $\mathcal{I}_0$ for further analysis in this subcase and denote $\mathcal{I}_0 = (G^2, \mathcal{T}, p)$; $G^2$ is the degree-reduced graph $G^1$.*

Let us summarize what Lemma 3.9 implies on the outcome of Branching Rule 4. We output at most $2^r rp$ instances and keep one instance for further analysis in each branch. Each output instance has strictly decreased potential, while $\phi((G^2, \mathcal{T}, p)) \leq \phi(G^1, \mathcal{T}, p))$. If $\mathcal{I}$ is a NO instance, all the generated instances—both the output and kept ones—are NO instances. If $\mathcal{I}$ is a YES instance, then it is possible that all the output instances are NO instances only if in the branch where the set $S$ is guessed correctly, the solution $Z$ is a solution to $(G^2, \mathcal{T}, p)$ as well. Moreover, as any solution to $(G^2, \mathcal{T}, p)$ is a solution to $\mathcal{I}$ as well by Lemma 3.8, in this case $Z$ is the lex-min solution to $(G^2, \mathcal{T}, p)$.

Let us now investigate more deeply the structure of the kept instances.

LEMMA 3.11. *In a branch, let $S$ be its label, $(G^1, \mathcal{T}, p)$ the instance on which Lemma 3.9 is invoked, and $(G^2, \mathcal{T}, p)$ the kept instance. For any $v \in V(G^1) = V(G^2)$ with $S(G, v) = S$, we have $S(G^1, v) = S$ and $S(G^2, v) \in \{\emptyset, S\}$.*

*Proof.* Note that the operation of bypassing a vertex $u$ does not change whether a vertex $v$ is reachable from a fixed source; thus $S(G, v) = S(G^1, v) = S$. By Lemma 3.8, we have $S(G^2, v) \subseteq S(G^1, v) = S$. Assume that $S(G^2, v) \neq \emptyset$, let $s_i \in S(G^2, v)$, let $P$ be a $s_i v$-path in $G^2$, and let $(s_i, w)$ be the first arc on this path. Since $G^2$ differs from $G^1$ only on the set of arcs incident to the set of sources, the subpath $P'$ of $P$ from $w$ to $v$ is present in $G^1$ as well. Therefore $S(G^1, w) \subseteq S(G^1, v) = S$. As $w$ was not bypassed by Reduction Rule 3, we have $S(G, w) = S$. Using again the fact that bypassing a vertex $u$ does not change whether $w$ is reachable from a fixed source, we have that $S(G^1, w) = S$. By Lemma 3.8, $S(G^1, w) = S(G^2, w) = S$. By the presence of $P'$ in $G^2$, we have $S \subseteq S(G^2, v)$. This finishes the proof of the lemma. ∎

Recall that if $\mathcal{I}$ is a YES instance and all instances output so far are NO instances, then in some subcase $S$ the set $Z$ is the lex-min solution to $(G^2, \mathcal{T}, p)$. In this case $Z$ does not contain any vertex from $V(G^2, \emptyset)$ and we can remove these vertices, as they are not contained in any $s_i t_i$-path for any $1 \leq i \leq r$.

REDUCTION RULE 5. *In each branch, let $S$ be its label and $(G^2, \mathcal{T}, p)$ the kept instance. As long as there exists a nonterminal vertex $v \in V(G^2)$ with $S(G^2, v) = \emptyset$, delete $v$. Denote the output instance by $(G^3, \mathcal{T}, p)$.*

Reduction Rule 5 does not interfere with any $s_i t_i$-paths, thus $\phi((G^3, \mathcal{T}, p)) = \phi((G^2, \mathcal{T}, p))$. Again, if $\mathcal{I}$ is a NO instance, all instances $(G^3, \mathcal{T}, p)$ are NO instances as well, and if $\mathcal{I}$ is a YES instance, but all output instances produced so far are NO instances, $Z$ is the lex-min solution to $(G^3, \mathcal{T}, p)$ in some branch $S$. Moreover, in $G^3$

each source has out-degree at most $rp$ and there is no vertex $v$ with $S(G^3, v) \subsetneq S$ (note that Reduction Rule 5 does not change reachability of a vertex from a fixed source). We apply the source shadow reduction to $(G^3, \mathcal{T}, p)$.

BRANCHING RULE 6. *In each branch, let $S$ be its label and $(G^3, \mathcal{T}, p)$ be the remaining instance. Invoke Lemma 2.11 on $(G^3, \mathcal{T}, p)$, obtaining a family $\mathcal{A}_S$. Branch into $|\mathcal{A}_S|$ subcases, labeled by pairs $(S, A)$ for $A \in \mathcal{A}_S$. In each subcase, obtain a graph $(G^4, \mathcal{T}, p)$ by bypassing (in arbitrary order) all vertices of $A \setminus N^+_{G^3}(T^s)$.*

Note that the graph $G^4$ does not depend on the order in which we bypass vertices of $A \setminus N^+_{G^3}(T^s)$. By Lemma 3.5, bypassing some vertices cannot turn a NO instance into a YES instance. Moreover, by Lemma 2.11, if $(G^3, \mathcal{T}, p)$ is a YES instance and $Z$ is the lex-min solution to $(G^3, \mathcal{T}, p)$, then there exists $A \in \mathcal{A}_S$ that contains all vertices of source shadows of $Z$, but no vertex of $Z$. Note that no out-neighbor of a source may be contained in a source shadow; therefore, $A \setminus N^+_{G^3}(T^s)$ contains all vertices of source shadows of $Z$ as well. We infer that in the branch $(S, A)$, $(G^4, \mathcal{T}, p)$ is a YES instance and, as bypassing a vertex only shrinks the set of solutions, $Z$ is still the lex-min solution to $(G^4, \mathcal{T}, p)$. Moreover, there are no source shadows of $Z$ in $(G^4, \mathcal{T}, p)$, and due to not bypassing $N^+_{G^3}(T^s)$ we have $|N^+_{G^4}(T^s)| \leq rp$.

At this point we have at most $2^{r+O(p^2)} \log |V(G)|$ subcases and at most $2^r rp$ already output instances. In each subcase, we have $\phi((G^4, \mathcal{T}, p)) \leq \phi((G^3, \mathcal{T}, p))$ by Lemma 3.5. The following observation is crucial for further branching.

LEMMA 3.12. *Take an instance $(G^4, \mathcal{T}, p)$ obtained in a branch labeled with $(S, A)$. Assume that $(G^4, \mathcal{T}, p)$ is a YES instance and let $Z$ be its lex-min solution. Moreover, assume that there are no source shadows of $Z$ in $(G^4, \mathcal{T}, p)$. Then the following holds: if there exists a vertex $v' \in (V(G, S) \cap V(G^4)) \setminus Z$ with $S(G^4 \setminus Z, v') \neq S$, then the first such vertex in the topological order $\preceq$ (denoted $v$) belongs to $N^+_{G^4}(T^s)$. Moreover, $v$ has at least one in-neighbor in $G^4$ that is not in $T^s$, and all such in-neighbors belong to $Z$.*

*Proof.* Let $v$ be as in the statement of the lemma. Assume there exists an in-neighbor $w$ of $v$ that is not in $T^s$ nor in $Z$. From the previous steps of the algorithm we infer that $S(G^4, v) = S(G^4, w) = S$. Moreover, $w \in V(G, S)$ as $v \in V(G, S)$: the vertex $v$ is reachable from $w$ in $G$ (possibly via vertices bypassed in Branching Rule 6), but all vertices $u$ with $S(G, u) \subsetneq S$ are bypassed in Reduction Rule 3. Since $w$ is earlier in $\preceq$ than $v$, from the minimality of $v$, we have $S(G^4 \setminus Z, w) = S$, a contradiction.

As there are no source shadows of $Z$ in $(G^4, \mathcal{T}, p)$, there exists $s_i \in S$ and an $s_i v$-path in $G^4$ that avoids $Z$. As $v$ has no in-neighbors outside $T^s$ and $Z$, this path consists of a single arc $(s_i, v)$ and $v \in N^+_{G^4}(T^s)$. Moreover, if all in-neighbors of $v$ in $G^4$ are sources, $S(G^4 \setminus Z, v) = S(G^4, v) = S$, a contradiction. ∎

BRANCHING RULE 7. *In each branch, let $(S, A)$ be its label and $(G^4, \mathcal{T}, p)$ the remaining instance. Output at most $rp$ instances $\mathcal{I}_v$, labeled by vertices $v \in N^+_{G^4}(T^s) \cap V(G, S)$ for which $N^-_{G^4}(v) \not\subseteq T^s$: the instance $\mathcal{I}_v$ is created from $(G^4, \mathcal{T}, p)$ by killing all nonterminal in-neighbors of $v$ and bypassing $v$. Moreover, create one remaining instance $(G^5, \mathcal{T}, p)$ as follows: delete from $G^4$ all arcs that have their ending vertices in $V(G, S) \cap V(G^4)$ and for each $v \in V(G, S) \cap V(G^4)$ and $s_i \in S$ add an arc $(s_i, v)$.*

By Lemmata 3.4 and 3.5, the output instances have strictly smaller potential than $(G^4, \mathcal{T}, p)$ and are NO instances if $(G^4, \mathcal{T}, p)$ is a NO instance. On the other hand, assume that $(G^4, \mathcal{T}, p)$ is a YES instance with lex-min solution $Z$ such that there are no source shadows of $Z$. If there exist vertices $v'$ and $v$ as in the statement of Lemma 3.12, then the instance $\mathcal{I}_v$ is computed and $Z \setminus N^-_{G^4}(v)$ (i.e., $Z$ without the killed vertices) is a solution to $\mathcal{I}_v$. Otherwise, we claim that $(G^5, \mathcal{T}, p)$ represents the remaining case.

LEMMA 3.13. *Let $(G^4, \mathcal{T}, p)$ be an instance obtained in the branch $(S, A)$.*

1. *$\phi((G^5, \mathcal{T}, p)) \leq \phi((G^4, \mathcal{T}, p))$.*
2. *Any multicut $Z$ in $(G^5, \mathcal{T}, p)$ is a multicut in $(G^4, \mathcal{T}, p)$ as well.*
3. *Assume additionally that $(G^4, \mathcal{T}, p)$ is a YES instance whose lex-min solution $Z$ satisfies the following properties: there are no source shadows of $Z$ and for each $v \in V(G, S) \cap V(G^4)$, either $v \in Z$ or $S(G^4 \setminus Z, v) = S$. Then $(G^5, \mathcal{T}, p)$ is a YES instance and $Z$ is its lex-min solution.*

*Proof.* For the first and second claims, consider an arbitrary $s_i t_i$-path $P$ in $(G^4, \mathcal{T}, p)$ for some $1 \leq i \leq r$. If $P$ does not contain any vertex from $V(G, S) \cap V(G^4)$, $P$ is present in $G^5$ as well and $Z$ intersects $P$. Otherwise, as $V(G, S) \cap V(G^4) \subseteq V(G^4, S)$, we have that $s_i \in S$. Let $v$ be the last (closest to $t_i$) vertex on $P$ that belongs to $V(G, S) \cap V(G^4)$. Note that $(s_i, v)$ is an arc of $G^5$; therefore a path $P'$ that starts with $(s_i, v)$ and then follows $P$ to $t_i$ is present in $G^5$. To prove the second point, note that as $Z$ is a multicut in $(G^5, \mathcal{T}, p)$, $Z$ intersects $P'$, and we infer that $Z$ intersects $P$. To prove the first point, note that the above reasoning shows that any set $\mathcal{P}$ of $s_i t_i$-paths in $G^4$ with pairwise disjoint interiors yields a family of the same number of $s_i t_i$-paths in $G^5$, again with pairwise disjoint interiors. Therefore, for any $1 \leq i \leq r$ we have $\mathrm{cut}_{G^4}(s_i, t_i) \leq \mathrm{cut}_{G^5}(s_i, t_i)$.

For the third claim, it is sufficient to prove that in the considered case the set $Z$ is a multicut in $(G^5, \mathcal{T}, p)$; its minimality follows from the second point. Consider an arbitrary $s_i t_i$-path $P'$ in $(G^5, \mathcal{T}, p)$ for some $1 \leq i \leq r$. Again, if $P'$ does not contain any vertex from $V(G, S) \cap V(G^4)$, $P'$ is present in $G^4$ as well and $Z$ intersects $P'$. Otherwise, $s_i \in S$ and $P'$ starts with an arc $(s_i, v)$ for some $v \in V(G, S) \cap V(G^4)$. Note that for any $v' \in V(G, S) \cap V(G^4)$, by construction we have $N^-_{G^5}(v') = S$. Therefore $v$ is the only vertex of $P'$ that belongs to $V(G, S) \cap V(G^4)$.

If $v \in Z$, $P'$ is intersected by $Z$ in $G^5$ and we are done. Otherwise, by the assumptions on $Z$, there exists an $s_i v$-path $P$ in $G^4 \setminus Z$. A concatenation of $P$ and $P'$ without the arc $(s_i, v)$ yields an $s_i t_i$-path in $G^4$. As $Z$ is a multicut in $(G^4, \mathcal{T}, p)$, we infer that $P'$ is intersected by $Z$ outside $V(G, S) \cap V(G^4)$ and the lemma is proved. □

The structure of $V(G, S) \cap V(G^5)$ is quite simple in $(G^5, \mathcal{T}, p)$. Recall that, if $\mathcal{I}$ is a YES instance, but no instance output so far is a YES instance, then in at least one branch $(S, A)$ we have that the lex-min solution $Z$ to $\mathcal{I}$ is the lex-min solution to $(G^5, \mathcal{T}, p)$ and $Z \cap V(G, S) \cap V(G^5) \neq \emptyset$. We would like to guess one vertex of $Z \cap V(G, S) \cap V(G^5)$. Although, $V(G, S) \cap V(G^5)$ may still be large, each vertex $v \in V(G, S) \cap V(G^5)$ has $N^-_{G^5}(v) = S$. Therefore we may limit the size of $V(G, S) \cap V(G^5)$ by applying once again the degree reduction branching.

BRANCHING RULE 8. *In each branch, let $(S, A)$ be its label and $(G^5, \mathcal{T}, p)$ the remaining instance. Apply Lemma 3.9 on $(G^5, \mathcal{T}, p)$, obtaining a sequence of instances $(\mathcal{I}_j)_{j=1}^d$ and the remaining instance $(G^6, \mathcal{T}, p)$, where $G^6$ is the degree-reduced graph $G^5$. Output all instances $\mathcal{I}_j$ for $1 \leq j \leq d$ and keep $(G^6, \mathcal{T}, p)$ for further analysis.*

By Lemma 3.9, if $(G^5, \mathcal{T}, p)$ is a NO instance, all the output instances as well as $(G^6, \mathcal{T}, p)$ are NO instances. Otherwise, if $(G^5, \mathcal{T}, p)$ is a YES instance with the lex-min solution $Z$, but the instances $\mathcal{I}_j$ are all NO instances, then $Z$ is the lex-min solution to $(G^6, \mathcal{T}, p)$.

Note that by Lemma 3.9, all output instances have potential strictly smaller than $\phi((G^5, \mathcal{T}, p))$, whereas $\phi((G^6, \mathcal{T}, p)) = \phi((G^5, \mathcal{T}, p))$. Moreover, applications of Branching Rule 8 in all subcases output at most $2^{r+O(p^2)} rp \log |V(G)|$ instances in total.

We are left with the final observation.

LEMMA 3.14. *In each subcase, let $(S, A)$ be its label and $(G^6, \mathcal{T}, p)$ the remaining instance. Then at most $rp$ vertices $v \in V(G, S) \cap V(G^6)$ have $S(G^6, v) \neq \emptyset$.*

*Proof.* Note that $V(G^4) = V(G^5) = V(G^6)$. Take $v \in V(G, S) \cap V(G^6)$. Recall that $N^-_{G^5}(v) = S$ and $G^6$ differs from $G^5$ only on the set of arcs incident to the sources, so $S(G^6, v) = N^-_{G^6}(v)$. The lemma follows from Lemma 3.8, claim 1. ∎

We may now perform once again Reduction Rule 5:

REDUCTION RULE 9. *In each branch, let $(S, A)$ be its label and $(G^6, \mathcal{T}, p)$ be the remaining instance. As long as there exists a nonterminal vertex $v \in V(G^6)$ with $S(G^6, v) = \emptyset$, delete $v$. Denote the output instance by $(G^7, \mathcal{T}, p)$.*

As in the case of Reduction Rule 5, $Z$ is the lex-min solution to $(G^6, \mathcal{T}, p)$ if and only if $Z$ is the lex-min solution to $(G^7, \mathcal{T}, p)$. Moreover, $\phi((G^6, \mathcal{T}, p)) = \phi((G^7, \mathcal{T}, p))$.

By Lemma 3.14, $|V(G, S) \cap V(G^7)| \leq rp$. Now we can perform final branching.

BRANCHING RULE 10. *In each subcase, let $(S, A)$ be its label and $(G^7, \mathcal{T}, p)$ the remaining instance. For each $v \in V(G, S) \cap V(G^7)$ output an instance $\mathcal{I}_v$ created from $(G^7, \mathcal{T}, p)$ by killing the vertex $v$.*

Note that if $V(G, S) \cap V(G^7) = \emptyset$, then this rule results in no branches created.

By Lemma 3.4, if $(G^7, \mathcal{T}, p)$ is a NO instance, so are the output instances $\mathcal{I}_v$. On the other hand, assume that $\mathcal{I}$ is a YES instance with the lex-min solution $Z$. Then in at least one subcase $(S, A)$, if no previously output instance is a YES instance, then the instance $(G^7, \mathcal{T}, p)$ is a YES instance, $Z$ is its lex-min solution, and $Z \cap V(G, S) \cap V(G^7) \neq \emptyset$. Then the instance $\mathcal{I}_v$ for any $v \in Z \cap V(G, S) \cap V(G^7)$ is a YES instance; in particular, $V(G, S) \cap V(G^7)$ is nonempty. To conclude the proof of Lemma 3.10 note that $\phi(\mathcal{I}_v) < \phi((G^7, \mathcal{T}, p))$ for each output instance $\mathcal{I}_v$.

We conclude with a short summary of the branching step:

1. Branching Rule 2 results in $2^r - 1$ subcases.
2. Branching Rule 4 outputs at most $rp$ instances and leaves one remaining instance in each subcase, less than $2^r rp$ output instances in total.
3. Branching Rule 6 results in $2^{O(p^2)} \log |V(G)|$ further subcases in each subcase; we have less than $2^{r+O(p^2)} \log |V(G)|$ subcases at this point.
4. Branching Rule 7 outputs at most $rp$ instances in each subcase and leaves one remaining instance, less than $2^{r+O(p^2)} rp \log |V(G)|$ output instances in total.
5. Branching Rule 8 outputs at most $rp$ instances in each subcase and leaves one remaining instance, less than $2^{r+O(p^2)} rp \log |V(G)|$ output instances in total.
6. Branching Rule 10 outputs at most $rp$ instances in each subcase and leaves no remaining instances, less than $2^{r+O(p^2)} rp \log |V(G)|$ output instances in total.

## 4. Lower bounds.

**4.1. W[1]-hardness of MULTICUT in DAGs parameterized by the size of the cutset.** The proof of Theorem 1.2 closely follows the lines of the proof of W[1]-hardness of the general case of MULTICUT in directed graphs of Marx and Razgon [13]. We simply need to replace the gadget $G_{i,j}$ (which is basically a long cycle) with its acyclic variant (depicted in Figure 2). For the sake of completeness, we include here a full proof.

*Proof of Theorem* 1.2. We show a polynomial-time reduction from the CLIQUE problem, known to be $W[1]$-hard. Let $(G, t)$ be a CLIQUE instance (i.e., we ask for a clique of size $t$ in the graph $G$). Denote $|V(G)| = n$ and $|E(G)| = m$.

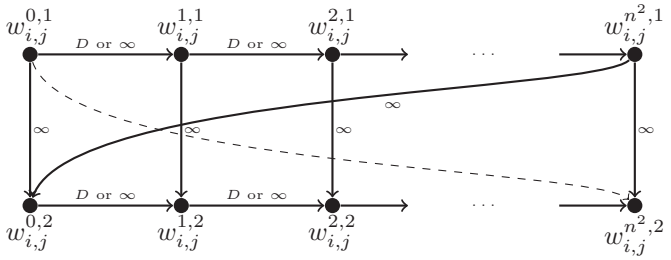


FIG. 2. *An acyclic variant of the gadget $G_{i,j}$. Dashed arc represents the terminal pair.*

Similarly as in [13], we prove W[1]-hardness of a weighted edge-deletion variant of MULTICUT IN DAGs. The edge- and node-deletion variants are easily seen to be equivalent (cf. [6]). In our construction we use three weights: light (one), heavy (polynomial in $t$), and infinite (which could be implemented as budget for cuts, $p$, plus one; $p$ will be polynomial in $t$). Therefore, all weights are polynomial in $t$, and the weighted variant can be easily reduced to the unweighted one by replacing arc $(u, v)$ of weight $\omega$ with $\omega$ $uv$-paths of length two.

For each ordered pair $(i, j)$, $1 \leq i, j \leq t$, $i \neq j$, we construct a gadget $G_{i,j}$ that has $2m$ states that encode a choice of one pair of adjacent vertices $(v_i, v_j)$ of the desired clique in $G$. We would like to ensure that the gadgets $G_{i,j}$ encode a clique $\{v_1, v_2, \ldots, v_t\}$ in $G$. As discussed in [13], it suffice to connect the gadgets in a way to ensure that

1. if $G_{i,j}$ represents $(v_i, v_j)$, then $G_{j,i}$ represents $(v_j, v_i)$;
2. if $G_{i,j}$ represents $(v_i, v_j)$ and $G_{i,j'}$ represents $(u_i, u_j)$, then $v_i = u_i$.

In particular, it follows from the above that if $G_{i,j}$ represents $(v_i, v_j)$ and $G_{i',j}$ represents $(u_i, u_j)$, then $v_j = u_j$.

Let $D = 2(t + 1)^2$ be the weight of a heavy arc. We set the budget for cuts as $p := 2t(t - 1)D + t(t + 1)/2$. Note that $p < 2t(t - 1)D + D$; thus we are allowed to cut only $2t(t - 1)$ heavy arcs.

We now describe the gadget $G_{i,j}$, depicted in Figure 2. Assume $V(G) = \{0, 1, \ldots, n - 1\}$ and let $\iota(x, y) = xn + y$ be a bijection from $V(G) \times V(G)$ to $\{0, 1, \ldots, n^2 - 1\}$. The gadget $G_{i,j}$ consists of $2n^2 + 2$ vertices $w_{i,j}^{s,\xi}$ for $0 \leq s \leq n^2$ and $\xi \in \{1, 2\}$. For $\xi \in \{1, 2\}$, $0 \leq s < n^2$, and $\iota^{-1}(s) = (x, y) \in V(G) \times V(G)$ we add an arc $(w_{i,j}^{s,\xi}, w_{i,j}^{s+1,\xi})$ of weight $D$ if $xy \in E(G)$ and $\infty$ otherwise. Moreover, we add an arc $(w_{i,j}^{n^2,1}, w_{i,j}^{0,2})$ and arcs $(w_{i,j}^{s,1}, w_{i,j}^{s,2})$ for $0 \leq s \leq n^2$, all of weight $\infty$. We define a terminal pair $(w_{i,j}^{0,1}, w_{i,j}^{n^2,2})$ in $G_{i,j}$.

Let us now analyze the gadget $G_{i,j}$. The terminals $(w_{i,j}^{0,1}, w_{i,j}^{n^2,2})$ are connected by two edge-disjoint paths $w_{i,j}^{0,1}, w_{i,j}^{1,1}, \ldots, w_{i,j}^{n^2,1}, w_{i,j}^{n^2,2}$ and $w_{i,j}^{0,1}, w_{i,j}^{0,2}, w_{i,j}^{1,2}, \ldots, w_{i,j}^{n^2,2}$. Therefore any solution needs to cut one edge $(w_{i,j}^{s,1}, w_{i,j}^{s+1,1})$ for some $0 \leq s < n^2$ and one edge $(w_{i,j}^{s',2}, w_{i,j}^{s'+1,2})$ for some $0 \leq s' < n^2$. As the cut budget $p$ allows us to cut only $2t(t - 1)$ heavy arcs (and no infinite ones), any solution cuts only the aforementioned two heavy arcs in each gadget $G_{i,j}$ and, apart from these, at most $t(t + 1)/2$ light arcs. Note that, moreover, we have that $s \leq s'$, as otherwise there remains a path $w_{i,j}^{0,1}, w_{i,j}^{1,1}, \ldots, w_{i,j}^{s'+1,1}, w_{i,j}^{s'+1,2}, w_{i,j}^{s'+2,2}, \ldots, w_{i,j}^{n^2,2}$. The index $s$ represents the choice made in gadget $G_{i,j}$, that is, if $\iota(x, y) = s$, then we say that $G_{i,j}$ represents the pair $(x, y)$. Note that if $G_{i,j}$ represents $s$ and $s_2 < s_1$, then $w_{i,j}^{s_2,2}$ is

reachable from $w_{i,j}^{s_1,1}$ if and only if $s_2 \leq s'$ and $s < s_1$; in particular, $w_{i,j}^{s,2}$ is reachable from $w_{i,j}^{s+1,1}$.

We now add connections between the gadgets to ensure the aforementioned properties, in a very similar fashion to [13].

In order to ensure the first property, for every $1 \leq i < j \leq t$ and for every ordered pair $(x, y) \in V(G) \times V(G)$, such that $xy \in E(G)$, we introduce two vertices $a_{i,j}^{(x,y)}$, $b_{i,j}^{(x,y)}$, an arc $(a_{i,j}^{(x,y)}, b_{i,j}^{(x,y)})$ of weight 1, two arcs $(w_{i,j}^{\iota(x,y),2}, a_{i,j}^{(x,y)})$, $(w_{j,i}^{\iota(y,x),2}, a_{i,j}^{(x,y)})$ of weight $\infty$, and two terminal pairs $(w_{i,j}^{\iota(x,y)+1,1}, b_{i,j}^{(x,y)})$ and $(w_{j,i}^{\iota(y,x)+1,1}, b_{i,j}^{(x,y)})$. Observe that if $G_{i,j}$ represents $(x, y)$, then $w_{i,j}^{\iota(x,y),2}$ is reachable from $w_{i,j}^{\iota(x,y)+1,1}$ and the arc $(a_{i,j}^{(x,y)}, b_{i,j}^{(x,y)})$ needs to be cut; similarly, if $G_{j,i}$ represents $(y', x')$, then $(a_{i,j}^{(x',y')}, b_{i,j}^{(x',y')})$ needs to be cut. If we are allowed to cut only one arc per choice of $1 \leq i < j \leq t$, then $x = x'$ and $y = y'$. Thus, if we have only $\binom{t}{2}$ cuts of light arcs for the connections introduced in this paragraph, the first property is ensured.

In order to ensure the second property, for each $1 \leq i \leq n$ and $x \in V(G)$ introduce two vertices $c_i^x$ and $d_i^x$ connected by an arc $(c_i^x, d_i^x)$ of weight 1. Furthermore, for every $1 \leq j \leq n$, $j \neq i$ we add an arc $(w_{i,j}^{\iota(x,0),2}, c_i^x)$ of weight $\infty$ and a terminal pair $(w_{i,j}^{\iota(x+1,0),1}, d_i^x)$. Note that if $G_{i,j}$ represents $(x, y)$, then $w_{i,j}^{\iota(x,0),2}$ is reachable from $w_{i,j}^{\iota(x+1,0),1}$ and the arc $(c_i^x, d_i^x)$ needs to be cut. If we are allowed only one cut per index $1 \leq i \leq n$ (i.e., $t$ cuts in total for connections introduced in this paragraph), then the second property would be satisfied. This concludes the description of the reduction.

To see that the constructed graph is acyclic, note that each gadget $G_{i,j}$ admits a topological order

$$w_{i,j}^{0,1}, w_{i,j}^{1,1}, \ldots, w_{i,j}^{n^2,1}, w_{i,j}^{0,2}, w_{i,j}^{1,2}, \ldots, w_{i,j}^{n^2,2}.$$

Moreover, all connections between the gadgets contain outgoing edges only; therefore a sequence that first contains all vertices of all gadgets (in the aforementioned order within each gadget), then all pairs $a_{i,j}^{(x,y)}, b_{i,j}^{(x,y)}$, and finally all pairs $c_i^x, d_i^x$ is a topological order of the constructed graph.

Let us now formally prove the equivalence. Let $\{v_1, v_2, \ldots, v_t\}$ be a set of vertices that induce a clique in $G$. Consider a set of arcs

$$\begin{aligned}
&\left\{(w_{i,j}^{\iota(v_i,v_j),\xi}, w_{i,j}^{\iota(v_i,v_j)+1,\xi}) : 1 \leq i, j \leq t, i \neq j, 1 \leq \xi \leq 2\right\} \\
&\quad \cup \left\{(a_{i,j}^{(v_i,v_j)}, b_{i,j}^{(v_i,v_j)} : 1 \leq i < j \leq t\right\} \\
&\quad \cup \{(c_i^{v_i}, d_i^{v_i}) : 1 \leq i \leq t\}
\end{aligned}$$

of weight exactly $p$. By the discussion on the gadgets $G_{i,j}$, the first group of arcs ensures that the terminal pair in each gadget $G_{i,j}$ is separated; note that the connections between the gadgets contain only arcs outgoing from the gadgets $G_{i,j}$, so all the paths between considered pairs of terminals have to be entirely contained in the corresponding gadget. Moreover, for any $0 \leq s_2 < s_1 \leq n^2$, in gadget $G_{i,j}$ the vertex $w_{i,j}^{s_2,2}$ is reachable from $w_{i,j}^{s_1,1}$ if and only if $s_2 \leq \iota(v_i, v_j) < s_1$. Therefore, if $(x, y) \neq (v_i, v_j)$, then the first group of arcs ensures that the terminal pair

$(w_{i,j}^{\iota(x,y)+1,1}, b_{i,j}^{(x,y)})$ (or $(w_{i,j}^{\iota(x,y)+1,1}, b_{j,i}^{(y,x)})$ if $i > j$) is separated; the second group separates the remaining pair for $(x, y) = (v_i, v_j)$. Similarly, if $x \neq v_i$, the first group of arcs ensures that the terminal pair $(w_{i,j}^{\iota(x+1,0),1}, d_i^x)$ is separated; the third group separates the remaining pair for $x = v_i$. We infer that the constructed graph admits a multicut of size $p$.

In the other direction, let $Z$ be a multicut in the constructed graph of size at most $p$. As discussed, $Z$ needs to contain exactly two arcs of weight $D$ from each gadget $G_{i,j}$ and each gadget $G_{i,j}$ represents some pair $(x, y)$. This leaves us with a budget of $t(t+1)/2 = \binom{t}{2} + t$ cuts of light arcs. We infer that we can spend one cut of an arc $(a_{i,j}^{(x,y)}, b_{i,j}^{(x,y)})$ per a pair $1 \leq i < j \leq t$ and only one cut of an arc $(c_i^x, d_i^x)$ per an index $1 \leq i \leq t$. Therefore, both properties of what the gadgets $G_{i,j}$ may represent are satisfied, so there are distinct vertices $v_1, v_2, \ldots, v_t$ such that gadget $G_{i,j}$ represents $(v_i, v_j)$. As in each gadget a finite weight was assigned only to an arc that corresponds to an edge in $G$, we obtain a clique of size $t$ in $G$. □

**4.2. NP-hardness of Skew Multicut.** In this section we prove Theorem 1.3.

*Proof of Theorem* 1.3. We provide a reduction from the NP-complete Max-Cut problem. Let us recall that the Max-Cut instance $(G, t)$ is an undirected graph together with an integer $t$ and we ask for a subset of vertices $X \subseteq V(G)$ such that there are at least $t$ edges of $G$ with exactly one endpoint in $X$. Denote $|V(G)| = n$ and $|E(G)| = m$.

We construct an equivalent Multicut in DAGs instance again in the arc-deletion setting; recall that the arc- and vertex-deletion variants are equivalent (cf. [6]). For clarity, we allow arcs to have weights: in our construction, we use infinite (of weight $p+1$, denoted $\infty$; $p$, the budget for cuts, will be polynomial in the size of $G$), heavy (of weight $D = 2m+1$), and light (of weight 1) arcs. As again the weights are polynomial in the size of $G$, we can easily reduce the weighted variant to the unweighted one by replacing an arc $uv$ of weight $\omega$ with $\omega$ $uv$-paths of length two.

We start a construction of an equivalent Skew Multicut instance by setting the cut budget $p = nD + 2m - t$ (as $p < nD + D$, we can delete only $n$ heavy edges) and by introducing two sources $s_1$, $s_2$ and two sinks $t_1$, $t_2$; recall that the set of terminal pairs is defined as $\mathcal{T} = \{(s_1, t_1), (s_1, t_2), (s_2, t_2)\}$.

For each vertex $v \in V(G)$ we introduce two vertices $a^v$ and $d^v$, as well as two arcs $(s_1, a^v)$ and $(d^v, t_2)$ of weight $D$ and an arc $(a^v, d^v)$ of weight $\infty$. We denote the path $s_1, a^v, d^v, t_2$ as $P_v$; note that any solution needs to cut one of the heavy arcs $(s_1, a^v)$ or $(d^v, t_2)$. As $p < nD + n$, each path $P_v$ is cut exactly once and the choice of the cut arc corresponds to the choice whether $v \in X$ or $v \in V(G) \setminus X$.

We now connect the paths $P_v$ in such a way that for an edge $uv \in E(G)$ we profit if the paths $P_u$ and $P_v$ are cut in a different manner. For each edge $uv \in E(G)$ we introduce four vertices $b_\alpha^{uv}$, $c_\alpha^{uv}$ for $\alpha \in \{1, 2\}$, two arcs $(b_1^{uv}, c_1^{uv})$ and $(b_2^{uv}, c_2^{uv})$ of weight 1, and eight arcs: $(s_2, b_\alpha^{uv})$, $(c_\alpha^{uv}, t_1)$ for $\alpha \in \{1, 2\}$ as well as $(a^u, b_1^{uv})$, $(a^v, b_2^{uv})$, $(c_2^{uv}, d^u)$, $(c_1^{uv}, d^v)$ of weight $\infty$. Note that the construction is symmetric with regard to $u$ and $v$ (i.e., changing the names of $u$ and $v$ results in changing the names of $b_1^{uv}$ and $c_1^{uv}$ with $b_2^{uv}$ and $c_2^{uv}$). The intuition behind this construction is as follows: if the paths $P_u$ and $P_v$ are cut in a different manner, we need to cut only one arc of weight 1 for the edge $uv$, and otherwise we need to cut both arcs. Part of the construction, with paths $P_u$, $P_v$ and the connection corresponding to the edge $uv$, is depicted in Figure 3.

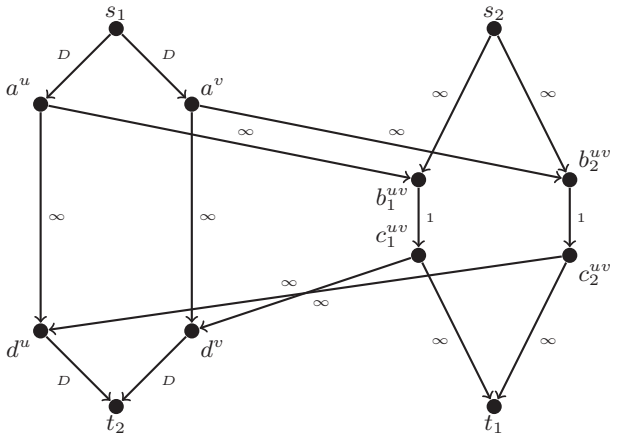


FIG. 3. *A part of the construction in the proof of Theorem* 1.3 *with paths* $P_u$ *and* $P_v$ *and their connection due to an edge* $uv$.

The following topological order of the constructed graph proves that we indeed construct an acyclic graph (within each set, we order the vertices arbitrarily):

$$\big\langle s_1, s_2, \{a^v : v \in V(G)\}, \{b_\alpha^{uv} : 1 \le \alpha \le 2, uv \in E(G)\},$$
$$\{c_\alpha^{uv} : 1 \le \alpha \le 2, uv \in E(G)\}, \{d^v : v \in V(G)\}, t_1, t_2 \big\rangle.$$

Let us now formally prove the equivalence of the input and output instances. Let $X \subseteq V(G)$ be such that there are at most $m - t$ edges in $E(G[X]) \cup E(G \setminus X)$. Consider the following set:

$$\begin{aligned} Z = {} & \{(s_1, a^v) : v \in X\} \cup \{(d^v, t_2) : v \in V(G) \setminus X\} \\ & \cup \{(b_1^{uv}, c_1^{uv}) : u \in V(G) \setminus X \vee v \in X\} \\ & \cup \{(b_2^{uv}, c_2^{uv}) : u \in X \vee v \in V(G) \setminus X\}. \end{aligned}$$

Intuitively, if $v \in X$, then we take the arc $(s_1, a^v)$ to the solution, and otherwise we take the arc $(d^v, t_2)$. If $u \in X$ and $v \in V(G) \setminus X$, then we only need to include the arc $(b_2^{uv}, c_2^{uv})$ in the solution; similarly, if $u \in V(G) \setminus X$ and $v \in X$, then we only need to include the arc $(b_1^{uv}, c_1^{uv})$. However, if $u \in X$ and $v \in X$, or $u \in V(G) \setminus X$ and $v \in V(G) \setminus X$, then both the arcs $(b_\alpha^{uv}, c_\alpha^{uv})$ for $\alpha \in \{1, 2\}$ need to be taken. As at least $t$ edges in $G$ have exactly one endpoint in $X$, we infer that the weight of $Z$ is at most $p$. It remains to check that $Z$ is a multicut in the constructed SKEW MULTICUT instance.

First, consider the terminal $s_1$. Its out-arc $(s_1, a^u)$ is not in $Z$ if and only if $u \in V(G) \setminus X$. The out-neighbors of $a^u$ are $d^u$ and $b_1^{uv}$, $b_2^{wu}$ for all edges $uv, wu \in E(G)$. From the construction of $Z$ we infer that $(d^u, t_2) \in Z$ and $(b_1^{uv}, c_1^{uv}), (b_2^{wu}, c_2^{wu}) \in Z$, thus $Z$ is a $s_1 - \{t_1, t_2\}$ separator. Symmetrically we show that $Z$ is an $\{s_1, s_2\} - t_2$ separator, and the constructed instance is a YES-instance to SKEW MULTICUT.

In the other direction, let $Z$ be a solution to the constructed instance of weight at most $p$. As discussed, $Z$ needs to contain exactly one heavy arc for each $v \in V(G)$, $(s_1, a^v)$ or $(d^v, t_2)$, and we are left with a budget of at most $2m - t$ light arcs. Let $X \subseteq V(G)$ be defined as the set of those vertices $v \in V(G)$ for which $(s_1, a^v) \in Z$.

Consider an edge $uv \in E(G)$. If $u \in X$, then $Z$ needs to intersect the path $s_2 - b_2^{uv} - c_2^{uv} - d^u - t_2$ in $(b_2^{uv}, c_2^{uv})$, and if $u \in V(G) \setminus X$, then $Z$ needs to intersect the path $s_1 - a^u - b_1^{uv} - c_1^{uv} - t_1$ in $(b_1^{uv}, c_1^{uv})$. Similarly, if $v \in X$, then $Z$ needs to intersect the path $s_2, b_1^{uv}, c_1^{uv}, d^v, t_2$ in $(b_1^{uv}, c_1^{uv})$, and if $v \in V(G) \setminus X$, then $Z$ needs to intersect the path $s_1, a^v, b_2^{uv}, c_2^{uv}, t_1$ in $(b_2^{uv}, c_2^{uv})$. We infer that for each edge $uv \in E(G)$, $Z$ needs to contain at least one arc $(b_\alpha^{uv}, c_\alpha^{uv})$ for $\alpha \in \{1, 2\}$ and both of them if $u, v \in X$ or $u, v \in V(G) \setminus X$. As $Z$ contains at most $2m - t$ light edges, $X$ is a solution to the Max-Cut instance $(G, t)$. ∎

**5. Conclusions.** In this paper we have proved that Multicut in DAGs, parameterized by the size of the cutset and the number of terminal pairs, admits an FPT algorithm working in time $2^{O(rp(r+p^2))}|V(G)|^{O(1)}$. This result is complemented by a proof that parameterization by the size of the cutset only is $W[1]$-hard. Thus, we obtain a full picture of the parameterized complexity of Multicut in DAGs.

A natural follow-up question is the complexity of Multicut in general directed graphs, parameterized by the size of the cutset and the number of terminal pairs. So far, parameterization by the size of the cutset only has been proved to be $W[1]$-hard by Marx and Razgon [13]. Although for two terminal pairs the problem can be easily reduced to Multiway Cut, to the best of our knowledge for three pairs its parameterized complexity remains open.

On the other hand, after designing an FPT algorithm one usually asks for existence of a polynomial kernel for the problem. The polynomial kernelization of many graph cut problems in directed graphs can be immediately refuted by the result of a superset of current authors that Multiway Cut, parameterized by the size of the cutset, does not admit a polynomial kernel in directed graphs even for two terminals, unless NP $\subseteq$ coNP/poly [7]. However, this does not cover the case of Multicut in DAGs. Hence, can a polynomial kernel for this problem also be proved to be implausible?

## REFERENCES


[1] C. Bentz, *On the hardness of finding near-optimal multicuts in directed acyclic graphs*, Theoret. Comput. Sci., 412 (2011), pp. 5325–5332.

[2] N. Bousquet, J. Daligault, and S. Thomassé, *Multicut is FPT*, in Proceedings of STOC, L. Fortnow and S. P. Vadhan, eds., ACM, 2011, pp. 459–468.

[3] J. Chen, Y. Liu, and S. Lu, *An improved parameterized algorithm for the minimum node multiway cut problem*, Algorithmica, 55 (2009), pp. 1–13.

[4] 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 (2008).

[5] R. H. Chitnis, M. Cygan, M. T. Hajiaghayi, and D. Marx, *Directed subset feedback vertex set is fixed-parameter tractable*, in ICALP 2012, Volume I, Lecture Notes in Comput. Sci. 7391, A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, eds., Springer, New York, 2012, pp. 230–241.

[6] R. H. Chitnis, M. T. Hajiaghayi, and D. Marx, *Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset*, SIAM J. Comput., 42 (2013), pp. 1674–1696.

[7] M. Cygan, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström, *Clique cover and graph separation: New incompressibility results*, Trans. Comput. Theory, 6 (2014).

[8] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk, *Subset feedback vertex set is fixed-parameter tractable*, SIAM J. Discrete Math., 27 (2013), pp. 290–309.

[9] M. R. Fellows, J. Guo, D. Marx, and S. Saurabh, *Data reduction and problem kernels*, Dagstuhl Reports, 2 (2012), pp. 26–50.

[10] S. Guillemot, *FPT algorithms for path-transversal and cycle-transversal problems*, Discrete Optim., 8 (2011), pp. 61–71.

[11] D. Lokshtanov and D. Marx, *Clustering with local restrictions*, Inf. Comput., 222 (2013), pp. 278–292.
[12] D. Marx, *Parameterized graph separation problems*, Theoret. Comput. Sci., 351 (2006), pp. 394–406.
[13] D. Marx and I. Razgon, *Fixed-parameter tractability of multicut parameterized by the size of the cutset*, SIAM J. Comput., 43 (2014), pp. 355–388.
[14] I. Razgon and B. O'Sullivan, *Almost* 2*-SAT is fixed-parameter tractable*, J. Comput. Systems Sci., 75 (2009), pp. 435–450.