

Exploring the complexity of layout parameters in tournaments and semi-complete digraphs*

Florian Barbero[†] Christophe Paul[‡] Michał Pilipczuk[§]

Abstract

A simple digraph is *semi-complete* if for any two of its vertices u and v , at least one of the arcs (u, v) and (v, u) is present. We study the complexity of computing two layout parameters of semi-complete digraphs: cutwidth and optimal linear arrangement (OLA). We prove that:

- Both parameters are NP-hard to compute and the known exact and parameterized algorithms for them have essentially optimal running times, assuming the Exponential Time Hypothesis.
- The cutwidth parameter admits a quadratic Turing kernel, whereas it does not admit any polynomial kernel unless $\text{NP} \subseteq \text{coNP}/\text{poly}$. By contrast, OLA admits a linear kernel.

These results essentially complete the complexity analysis of computing cutwidth and OLA on semi-complete digraphs. Our techniques can be also used to analyze the sizes of minimal obstructions for having small cutwidth under the induced subdigraph relation.

Contents

1	Introduction	2
2	Preliminaries and basic results	4
3	Polynomial-time and approximation algorithms	6
4	Kernelization aspects of CUTWIDTH and OLA	9
4.1	On polynomial kernels	9
4.2	A Turing kernel for the CUTWIDTH problem	10
5	Cutwidth-minimal semi-complete digraphs	17
5.1	Upper bound for c -cutwidth-minimal tournaments	17
5.2	Upper bound for c -cutwidth-minimal semi-complete digraphs	19
5.3	Algorithmic applications	20
6	Lower bounds	24
7	Parameterization by the number of pure vertices	31
8	Conclusions	33

*The research of F. Barbero and C. Paul is supported by the DE-MO-GRAPH project ANR-16-CE40-0028. The research of Mi. Pilipczuk is supported by Polish National Science Centre grant UMO-2013/11/D/ST6/03073. Mi. Pilipczuk is also supported by the Foundation for Polish Science via the START stipend programme.

[†]LIRMM, Université de Montpellier, France, florian.barbero@lirmm.fr

[‡]LIRMM, CNRS, Université de Montpellier, France, christophe.paul@lirmm.fr

[§]University of Warsaw, Poland, michal.pilipczuk@mimuw.edu.pl

1 Introduction

A directed graph (digraph) is *simple* if it does not contain a self-loop or multiple arcs with the same head and tail. A simple digraph is *semi-complete* if for any pair of its vertices u and v , at least one of the arcs (u, v) or (v, u) is present. If moreover exactly one of them is present for each pair u, v , then a semi-complete digraph is called a *tournament*. Tournaments and semi-complete digraphs form rich and interesting subclasses of directed graphs; we refer to the book of Bang-Jensen and Gutin [2] for an overview.

We study two layout parameters for tournaments and semi-complete digraphs: cutwidth and optimal linear arrangement (OLA). Suppose π is an ordering of the vertices of a digraph D . With each prefix of π we associate a *cut* defined as the set of arcs with head in the prefix and tail outside of it. The *width* of π is defined as the maximum size among the cuts associated with the prefixes of π . The *cutwidth* of D , denoted $\text{ctw}(D)$, is the minimum width among orderings of the vertex set of D . Optimal linear arrangement (OLA) is defined similarly, but when defining the width of π , called in this context the *cost* of π , we take the sum of the cutsizes associated with prefixes, instead of the maximum. Then the OLA-cost of a digraph D , denoted $\text{OLA}(D)$, is the minimum cost among vertex orderings of D .

Known results. The study of cutwidth in the context of tournaments and semi-complete digraphs started with the work of Chudnovsky, Fradkin, and Seymour [5, 6, 14], who identified this layout parameter as the accurate dual notion to immersions in semi-complete digraphs. In particular, it is known that excluding a fixed digraph as an immersion yields a constant upper bound on the cutwidth of a semi-complete digraph [5, 20]. Due to this connection, cutwidth played a pivotal role in the proof of Chudnovsky and Seymour that the immersion order is a well quasi-order on tournaments [6].

The algorithmic properties of cutwidth were preliminarily investigated by Chudnovsky, Fradkin, and Seymour [5, 14, 13]. In Fradkin's PhD thesis [13], several results on the tractability of computing the cutwidth are presented. In particular, it is shown that the cutwidth of a tournament can be computed optimally by just sorting vertices according to their outdegrees, whereas in semi-complete digraphs a similar approach yields a polynomial-time 2-approximation algorithm. The problem becomes NP-hard on super-tournaments, that is, when multiple parallel arcs are allowed. Later, the third author together with Fomin proposed a parameterized algorithm for computing the cutwidth of a semi-complete digraph with running time $2^{\mathcal{O}(\sqrt{k \log k})} \cdot n^2$ [11, 21], where n is the number of vertices and k is the target width. Using the same techniques, OLA in semi-complete digraphs can be solved in time $2^{\mathcal{O}(k^{1/3} \sqrt{\log k})} \cdot n^2$ [11, 21], where k is the target cost. It was left open whether the running times of these parameterized algorithms are optimal [21]. In fact, even settling the NP-hardness of computing cutwidth and OLA in semi-complete digraphs was open [13, 21].

Our contribution. We study two aspects of the computational complexity of computing cutwidth and OLA of semi-complete digraphs: optimality of parameterized algorithms and kernelization. First, we prove that these problems are NP-hard and we provide almost tight lower bounds for the running times of algorithms solving them, based on the Exponential Time Hypothesis

(ETH). More precisely, assuming ETH we prove that the cutwidth of a semi-complete digraph cannot be computed in time $2^{o(n)}$ nor in time $2^{o(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$. The same arguments yields $2^{o(n)}$ and $2^{o(k^{1/3})} \cdot n^{\mathcal{O}(1)}$ lower bounds on the time to compute the OLA-cost of a semi-complete digraph. (See Theorem 14 in Section 6 for a precise statement.) It follows that the known parameterized algorithms of Fomin and Pilipczuk [11] are optimal under ETH, up to $\sqrt{\log k}$ factor in the exponent. Note that both cutwidth and OLA-cost can be computed in time $2^n \cdot n^{\mathcal{O}(1)}$ using standard dynamic programming on subsets, so we obtain tight lower bounds also for exact exponential-time algorithms. Interestingly, the lower bounds for both problems are based on the same reduction.

Next we turn our attention to the kernelization complexity of computing the two layout parameters (see Section 4). We first observe that, when respectively parameterized by the target cost and the target width, the two problems behave differently. As far as OLA is concerned, we prove that there is a polynomial-time algorithm that, given an arbitrary digraph D and a positive integer c , either correctly concludes that $\text{OLA}(D) > c$, or finds a digraph D' on at most $2c$ vertices such that $\text{OLA}(D') = \text{OLA}(D)$. That is, computing the OLA-cost of an arbitrary digraphs admits a kernel of size linear in the target cost (see Theorem 4). On the other hand, a simple AND-composition [9, 7] shows that, under the assumption that $\text{NP} \not\subseteq \text{coNP}/\text{poly}$, such a polynomial size kernel is unlikely to exist for the problem of computing the cutwidth of a semi-complete digraph (see Theorem 5).

Surprisingly, pre-processing for computing the cutwidth of a semi-complete digraph turns out to be efficient if we consider the alternative notion of kernelization called *Turing kernelization*. In this framework, which has also been studied intensively in the literature (cf. the discussion in [7]), it is not required that the instance at hand is reduced to one equivalent small instance, but rather that the whole problem can be solved in polynomial time assuming oracle access to an algorithm solving instances of size bounded by a function of the parameter.

We design a polynomial-time algorithm that given a semi-complete digraph D and integer c , either correctly concludes that $\text{ctw}(D) > c$ or outputs a list of at most n induced subdigraphs D_1, \dots, D_ℓ of D , each with at most $\mathcal{O}(c^2)$ vertices, such that $\text{ctw}(D) \leq c$ if and only if $\text{ctw}(D_i) \leq c$ for each $i \in \{1, 2, \dots, \ell\}$ (see Theorem 7). Observe that this algorithm is a so-called AND-*Turing kernel*, meaning that it just computes the output list without any oracle calls, and the answer to the input instance is the conjunction of the answers to the output small instances. This places the problem of computing the cutwidth of a semi-complete digraph among very few known examples of natural problems where classic and Turing kernelization have different computational power [3, 19, 15, 22, 25]. Moreover, this is the first known to us polynomial AND-Turing kernel for a natural problem: examples of Turing kernelization known in the literature are either OR-Turing kernels [3, 15, 22], or adaptative kernels that fully exploit the oracle model [19, 25]. As separating classic and Turing kernelization is arguably one of the most important complexity-theoretical open problems within parameterized complexity [8, 17, 7], we find this new example intriguing.

The proof of the Turing kernel relies on the notion of a *lean ordering*; see e.g. [16, 24]. Intuitively, a vertex ordering is lean if it is tight with respect to cut-flow duality: there are systems of arc-disjoint paths which certify that cutsizes along the ordering cannot be improved. Lean orderings and decompositions are commonly used in the analysis of obstructions for various

width notions, as well as for proving well quasi-order results. In particular, the concept of a lean ordering for cutwidth of digraphs was used by Chudnovsky and Seymour in their proof that the immersion order is a well quasi-order on tournaments [6].

As a byproduct of our approach to proving Theorem 7, we obtain also polynomial upper bounds on the sizes of minimal obstructions to having small cutwidth. For a positive integer c , a digraph D is called *c-cutwidth-minimal* if the cutwidth of D is at least c , but the cutwidth of every proper induced subdigraph of D is smaller than c . We show that every c -cutwidth-minimal semi-complete digraph has at most $\mathcal{O}(c^2)$ vertices (see Theorem 9) and that every c -cutwidth-minimal tournament has at most $2c + 2\lceil\sqrt{2c}\rceil + 1$ vertices 8. See Section 5. We remark that from the well-quasi order result of Chudnovsky and Seymour [6], it follows the number of minimal *immersion* obstructions for tournaments of cutwidth at most c is finite. However, this holds only for tournaments, yields a non-explicit upper bound on obstruction sizes, and applies to immersion and not induced subdigraph obstructions. As discussed in Section 5, these bounds have direct algorithmic applications for parameterized graph modification problems related to cutwidth, e.g., c -CUTWIDTH VERTEX DELETION: remove at most k vertices from a given digraph to obtain a digraph of cutwidth at most c .

Organization. In Section 2 we establish basic notation and recall some background from complexity theory, concerning kernelization and the Exponential Time Hypothesis. In Section 3 we give polynomial-time algorithms for computing the cutwidth and the OLA-cost of a tournament, as well as 2-approximation algorithms for these parameters on semi-complete digraphs. Section 4 is devoted to the kernelization complexity. We first describe the complexity of computing the cutwidth and the OLA-cost of a semi-complete digraph under the classic notion of kernelization, and then we give a quadratic Turing kernel for the cutwidth case. In Section 5 we utilize the tools developed in the previous sections to give upper bounds on the sizes of c -cutwidth-minimal tournaments and semi-complete digraphs. Section 6 focuses on lower bounds under ETH: we prove that in the semi-complete setting, both computing the cutwidth and the OLA-cost are NP-complete problems, and the running times of known algorithms for them are essentially tight under ETH. The shape of the lower bound construction suggests that the problem may be tractable if almost all vertices are not incident to any symmetric arcs (two oppositely oriented arcs between the same pair of vertices). Indeed, in Section 7 we present an appropriate parameterization that captures this scenario, and prove a fixed-parameter tractability result for it. Section 8 concludes the paper by giving some final remarks.

2 Preliminaries and basic results

Notations. We use standard graph notation for digraphs. For a digraph D , the vertex and edge sets of D are denoted by $V(D)$ and $E(D)$, respectively. For $X, Y \subseteq V(D)$, we denote $E(X, Y) = \{(u, v) \in E(D) : u \in X, v \in Y\}$. The *subdigraph induced* by X is the digraph $D[X] = (X, E(X, X))$. Note that an induced subdigraph of a tournament is also a tournament, likewise for semi-complete digraphs. The *inneighborhood* of a vertex u is $N_D^-(u) = \{v \in V(D) : (v, u) \in E(D)\}$, a vertex $v \in N_D^-(u)$ being an *inneighbour* of u . The *indegree* of u is $d_D^-(u) = |N_D^-(u)|$. We may drop the subscript when it is clear from the context. We define

similarly the *outneighborhood* $N^+(u)$ and the *outdegree* $d^+(u)$. All digraphs considered in this paper are simple, i.e., they do not contain a self-loop or multiple arcs with the same head and tail. For definitions of tournaments and semi-complete digraphs, see the first paragraph of Section 1. If present in a digraph, the arcs (u, v) and (v, u) are called *symmetric arcs*.

For two integers $p \leq p'$, let $[p, p'] \subseteq \mathbb{Z}$ be the set of integers between p and p' . If $p < p'$, we set $[p', p] = \emptyset$ by convention. A *vertex ordering* of a digraph D is a bijective mapping $\pi: V(D) \rightarrow [1, n]$, where $n = |V(D)|$. A vertex $u \in V(D)$ is *at position i in π* if $\pi(u) = i$. We denote this unique vertex by π_i . The *prefix* of length i of π is $\pi_{\leq i} = \{\pi_j: j \in [1, i]\}$; we set $\pi_{\leq i} = \emptyset$ when $i \leq 0$, and $\pi_{\leq i} = V(D)$ when $n \leq i$. We extend this notation to prefixes and suffixes of orderings naturally, e.g., $\pi_{> i} = V(D) \setminus \pi_{\leq i}$ is the set of the last $n - i$ vertices in π . The notions of restriction of an ordering to a subset of vertices and of concatenation of orderings are defined naturally.

An arc $(\pi_i, \pi_j) \in E(D)$ is a *feedback arc for π* if $i > j$, that is, if π_i is after π_j in π . Given a digraph $D = (V, E)$, an ordering π of V and an integer i , we define the *cut* E_π^i as the set of feedback arcs $E(\pi_{> i}, \pi_{\leq i})$. The tuple $\text{cuts}\langle D, \pi \rangle = (|E_\pi^0|, |E_\pi^1|, \dots, |E_\pi^n|)$ is called the *cut vector* of π , and we denote $\text{cuts}\langle D, \pi \rangle(i) = |E_\pi^i|$. Let \preceq be the product order on tuples: for n -tuples A, B , we have $A \preceq B$ iff $A(i) \leq B(i)$ for all $i \in [0, n]$. We define $A \prec B$ as $A \preceq B$ and $A \neq B$. We say that a vertex ordering π is *minimum for D* if for all vertex orderings π' of D we have $\text{cuts}\langle D, \pi \rangle \preceq \text{cuts}\langle D, \pi' \rangle$. Note that a minimum vertex ordering may not exist.

The *width* of a vertex ordering π of a digraph D , denoted $\text{ctw}(D, \pi)$, is equal to $\max\{\text{cuts}\langle D, \pi \rangle\}$, where \max on a tuple yields the largest coordinate. The *cutwidth* of D , denoted $\text{ctw}(D)$, is the minimum width among vertex orderings of D . Similarly, the *cost* of π , denoted $\text{OLA}(D, \pi)$, is equal to $\sum\{\text{cuts}\langle D, \pi \rangle\}$, where \sum on a tuple yields the sum of coordinates. This is equivalent to summing $|i - j|$ for all feedback arcs (π_j, π_i) in π . The *OLA-cost* of D , denoted $\text{OLA}(D)$, is the minimum cost among vertex orderings of D . A vertex ordering π of D satisfying $\text{ctw}(D) = \text{ctw}(D, \pi)$, or $\text{OLA}(D) = \text{OLA}(D, \pi)$, is respectively called *ctw-optimal* or *OLA-optimal* for D . Note that a minimum ordering for D , if existent, is always *ctw-optimal* and *OLA-optimal* for D .

Kernelization. A *kernelization algorithm* (or *kernel*, for short) is a polynomial-time algorithm that given some instance of a parameterized problem, returns an equivalent instance whose size is bounded by a computable function of the input parameter; this function is called the *size* of the kernel. We are mostly interested in finding polynomial kernels, as admitting a kernel of any computable size is equivalent to fixed-parameter tractability of the problem [8, 7].

Based on the complexity hypothesis that $\text{NP} \not\subseteq \text{coNP}/\text{poly}$, various parameterized reduction and composition techniques allow to rule out the existence of a polynomial-size kernel. We refer to e.g. [7] for an overview of this methodology. The concept of *AND-composition* is one of these techniques. For a parameterized problem Π (see [8, 7] for basic definitions of parameterized complexity), an *AND-composition* is a polynomial-time algorithm that a sequence of instances $(I_1, k), (I_2, k), \dots, (I_t, k)$, all with the same parameter k , outputs a single instance (I^*, k^*) , such that the following holds:

- $(I^*, k^*) \in \Pi$ if and only if $(I_i, k) \in \Pi$ for all $1 \leq i \leq t$; and

- $k^* \leq q(k)$ for some polynomial q .

Theorem 1 ([9]). *Let Π be an NP-hard parameterized problem for which there exists an AND-composition. Then, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$, Π does not admit a polynomial kernel.*

The notion of *Turing kernelization* has been introduced as an alternative to classic kernelization (cf. the discussion in [7]). The idea is to relax the notion of efficient pre-processing to allow producing multiple instances of small size, so that we can obtain positive results for problems which do not admit polynomial-size kernels in the strict sense. In this framework, it is not required that the instance at hand is reduced to one equivalent small instance, but rather that the whole problem can be solved in polynomial time assuming oracle access to an algorithm solving instances of size bounded by a function of the parameter. More precisely, a Turing kernel of size $q(k)$ for a parameterized problem Π is an algorithm that, given an instance (I, k) , resolves whether $(I, k) \in \Pi$ in polynomial time when given access to an oracle that resolves belonging to Π for instances of size at most $q(k)$. Each oracle call is counted as a single step of the algorithm. As with classic kernelization, we typically assume that $q(k)$ is computable, and in practice we are looking for *polynomial Turing kernels* where $q(k)$ is a polynomial.

Exponential-Time Hypothesis. The *Exponential Time Hypothesis* (ETH) of Impagliazzo et al. [18] states that for some constant $\alpha > 0$, there is no algorithm for 3SAT that would run in time $2^{\alpha \cdot n} \cdot (n + m)^{\mathcal{O}(1)}$, where n and m are the numbers of variables and clauses of the input formula, respectively. Using the *Sparsification Lemma* [18] one can show that under ETH, there is a constant $\alpha > 0$ such that 3SAT cannot be solved in time $2^{\alpha \cdot m} \cdot (n + m)^{\mathcal{O}(1)}$. In this work we use the NAE-3SAT problem (for Not-All-Equal), which is a variant of 3SAT where a clause is considered satisfied only when at least one, but not all of its literals are satisfied. Schaefer [23] gave a linear reduction from 3SAT to NAE-3SAT, which immediately yields:

Corollary 1. *Unless ETH fails, NAE-3SAT cannot be solved in time $2^{o(m)} \cdot (n + m)^{\mathcal{O}(1)}$, where n and m are the numbers of variables and clauses of the input formula, respectively.*

3 Polynomial-time and approximation algorithms

The starting point of our study is the approach used in the earlier works by Fradkin [13] and by the third author [21, 20], namely to sort the vertices of the given semi-complete digraph according to non-decreasing indegrees, and argue that this ordering has to resemble an optimum one. As shown by Fradkin [13], this statement may be made precise for tournaments: any indegree ordering has optimum cutwidth. A slight modification enabled Fradkin [13] to establish a polynomial-time 2-approximation for semi-complete digraphs. We choose to include the proofs of these two results (see Theorems 2 and 3 below) for several reasons. First, the applicability of the approach to OLA is a new contribution of this work. Moreover, the fine understanding of optimum orderings is a basic tool needed in the proofs of our main results. Finally, the abovementioned results of Fradkin [13] were communicated only in her PhD thesis and, to the best of our knowledge, were neither included in any published work, nor we have found any reference to them. We believe that these fundamental observations deserve a better publicity.

The core idea is to work in the more general setting of *fractional tournaments*, a linear relaxations of tournaments. Formally, a *fractional tournament* is a pair $T = (V, \omega)$, where V is a finite vertex set and $\omega: V^2 \rightarrow \mathbb{R}_{\geq 0}$ is a weight function that satisfies the following properties: $\omega(u, u) = 0$ for all $u \in V$, and $\omega(u, v) + \omega(v, u) = 1$ for all pairs of different vertices u, v . Thus, by requiring the weights to be integral we recover the original definition of a tournament. We extend the notation for digraphs to fractional tournaments as follows. For $X, Y \subseteq V$ we define $\omega(X, Y) = \sum_{x \in X, y \in Y} \omega(x, y)$, and for $u \in V$ we define $\omega^-(u) = \omega(V, \{u\})$ and $\omega^+(u) = \omega(\{u\}, V)$. The notions of (minimum) vertex orderings, cut vectors, cutwidth, and OLA-cost are extended naturally: the cardinality of any cut $E(X, Y)$ is replaced by the sum of weights $\omega(X, Y)$.

Suppose $T = (V, \omega)$ is a fractional tournament. We say that a vertex ordering π of T is *sorted* if for any pair of different vertices u and v , if $\omega^-(u) < \omega^-(v)$, then $\pi(u) < \pi(v)$; in other words, the vertices are sorted according to their indegrees. The following lemma extends the sorting arguments of Fradkin [13], and encapsulates the essence of our approach.

Lemma 1. *A vertex ordering of a fractional tournament is minimum iff it is sorted.*

Proof. Let $T = (V, \omega)$ be a fractional tournament. By definition, a vertex ordering σ of T is minimum if and only if for each vertex ordering π and each $i \in [0, |V|]$, the following inequality (\diamond) holds: $\text{cuts}\langle T, \sigma \rangle(i) \leq \text{cuts}\langle T, \pi \rangle(i)$. Observe that, since T is a fractional tournament, for any vertex ordering π of T and position $i \in [0, |V|]$ we have that

$$\begin{aligned} \text{cuts}\langle T, \pi \rangle(i) &= \omega(\pi_{>i}, \pi_{\leq i}) = \omega(V, \pi_{\leq i}) - \omega(\pi_{\leq i}, \pi_{\leq i}), \text{ and} \\ \omega(\pi_{\leq i}, \pi_{\leq i}) &= \sum_{u \in \pi_{\leq i}} \omega(u, u) + \sum_{\substack{\{u, v\} \subseteq \pi_{\leq i}, \\ u \neq v}} (\omega(u, v) + \omega(v, u)) = \binom{i}{2}. \end{aligned}$$

Hence, the inequality (\diamond) is equivalent to $\omega(V, \sigma_{\leq i}) \leq \omega(V, \pi_{\leq i})$. It is clear that the vertex ordering σ minimizes $\omega(V, \sigma_{\leq i})$ only when $\sigma_{\leq i}$ consists of i vertices of T with the smallest indegrees ω^- . But the latter condition holds for all $i \in [0, |V|]$ if and only if σ is sorted. \square

Theorem 2. *The cutwidth and OLA of a tournament can be computed in polynomial time.*

The proof of Theorem 2, even in the more general setting of fractional tournaments, is now immediate. We just sort the vertices according to their indegrees ω^- . By Lemma 1, the obtained ordering is minimum, hence it is both *ctw*-optimal and *OLA*-optimal.

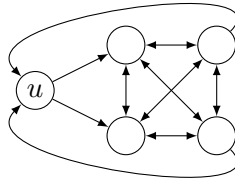


Fig. 1: A semi-complete digraph without any minimum ordering. Since u has both the minimum indegree and the minimum outdegree, it minimizes the first or last cut depending whether it is the first or last vertex of an ordering.

Observe that Lemma 1 cannot be generalized to the semi-complete setting, as there are semi-complete digraphs that do not admit any minimum ordering; see Figure 1 for an example. However, relaxing a given semi-complete digraph to a fractional tournament yields a 2-approximation algorithm for general semi-complete digraphs.

Precisely, for a semi-complete digraph D , consider its *relaxation* T_D which is a fractional tournament on the vertex set $V(D)$, where for every pair of different vertices u and v , we put:

- $\omega(u, v) = 1$ and $\omega(v, u) = 0$, when (u, v) is present in D but (v, u) is not present; and
- $\omega(u, v) = \omega(v, u) = 1/2$, when (u, v) and (v, u) is a pair of symmetric arcs in D .

We put $\omega(u, u) = 0$ for every vertex u , thus T_D is indeed a fractional tournament. Observe that for any pair of vertices u, v , we have $|E(\{u\}, \{v\})|/2 \leq \omega_{T_D}(\{u\}, \{v\}) \leq |E(\{u\}, \{v\})|$. Therefore, for every vertex ordering π of D and every index $i \in [0, n]$, it holds that

$$\text{cuts}\langle D, \pi \rangle(i)/2 \leq \text{cuts}\langle T_D, \pi \rangle(i) \leq \text{cuts}\langle D, \pi \rangle(i).$$

In particular we have $\text{ctw}(D)/2 \leq \text{ctw}(T_D) \leq \text{ctw}(D)$ and $\text{OLA}(D)/2 \leq \text{OLA}(T_D) \leq \text{OLA}(D)$. As a consequence, we have:

Theorem 3. *There exists a polynomial-time algorithm that given a semi-complete digraph D , outputs an ordering of its vertices of width, and respectively cost, upper bounded by twice the cutwidth, respectively OLA-cost, of D .*

Proof. Given a semi-complete digraph D , we compute its relaxation T_D and we sort the vertices according to their indegrees in T_D , obtaining a vertex ordering π . Lemma 1 ensures us that

$$\text{ctw}(T_D, \pi) = \text{ctw}(T_D) \quad \text{and} \quad \text{OLA}(T_D, \pi) = \text{OLA}(T_D).$$

On the other hand, by inequalities

$$\text{ctw}(D)/2 \leq \text{ctw}(T_D) \leq \text{ctw}(D) \quad \text{and} \quad \text{OLA}(D)/2 \leq \text{OLA}(T_D) \leq \text{OLA}(D)$$

we have that

$$\begin{aligned} \text{ctw}(D, \pi)/2 &\leq \text{ctw}(T_D, \pi), & \text{OLA}(D, \pi)/2 &\leq \text{OLA}(T_D, \pi), \\ \text{ctw}(T_D) &\leq \text{ctw}(D), & \text{OLA}(T_D) &\leq \text{OLA}(D). \end{aligned}$$

By combining these together, we obtain the required upper bounds:

$$\text{ctw}(D, \pi) \leq 2\text{ctw}(D) \quad \text{and} \quad \text{OLA}(D, \pi) \leq 2\text{OLA}(D),$$

which means that the ordering π can be reported by the algorithm. \square

Finally, we remark that Theorem 3 can be generalized in several directions. First, here we considered parameters cutwidth and OLA-cost, which are defined by taking the maximum and the sum over the cut vector corresponding to an ordering. However, we used only a few basic properties of the max and sum functions; the whole reasoning would go through for any function over the cut vectors that is monotone with respect to the \preceq order, and scales by α when the cut vector is multiplied by α . Second, instead of semi-complete digraphs, we could consider

weighted semi-complete digraphs defined as follows: we have a weight function $\omega: V^2 \rightarrow \mathbb{R}_{\geq 0}$, where we require that $\omega(u, u) = 0$ for every vertex u , and $\omega(u, v) + \omega(v, u) > 0$ for every pair of different vertices u and v . Such a weighted semi-complete digraph can be rescaled to a fractional tournament by taking the normalized weight function $\omega'(u, v) = \frac{\omega(u, v)}{\omega(u, v) + \omega(v, u)}$. Then, the same reasoning as above yields an approximation algorithm for the cutwidth and the OLA-cost with approximation factor $\frac{\omega_D^{\max}}{\omega_D^{\min}}$, where

$$\omega_D^{\max} = \max_{\substack{u, v \in V(D) \\ u \neq v}} (\omega(u, v) + \omega(v, u)) \quad \text{and} \quad \omega_D^{\min} = \min_{\substack{u, v \in V(D) \\ u \neq v}} (\omega(u, v) + \omega(v, u)).$$

4 Kernelization aspects of CUTWIDTH and OLA

4.1 On polynomial kernels

As announced in the introduction, the CUTWIDTH and OLA problems, when respectively parameterized by the target width and target cost, behave differently with respect to the kernelization paradigm. As we shall see next, the problem of computing the OLA-cost admits a linear kernel in arbitrary digraphs, while computing the CUTWIDTH of semi-complete digraphs is unlikely to admit a polynomial size kernel. The proofs of these two results, stated as Theorems 4 and 5 below, directly follow from the understanding of the contribution of strongly connected components in optimal orderings. On one side, the contribution to the OLA-cost of each strongly connected component is at least linear in its size, implying a linear kernel for this parameter. On the other side, we can observe that the cutwidth of digraph is the maximum over the cutwidth of its strongly connected components, which implies that, like many other width parameters, cutwidth is an *AND-composable* parameter [9, 7].

Theorem 4. *The OLA problem in arbitrary digraphs, parameterized by the target cost k , admits a kernel with at most $2k$ vertices.*

Proof. Let D be a digraph. As the set of strongly connected components of a digraph naturally defines an acyclic digraph, we can choose an ordering C_1, \dots, C_p of the strongly connected components so that whenever $u \in C_i$ and $v \in C_j$ for some $i < j$, then the arc (v, u) is not present in D (such an ordering is given by a topological ordering of the DAG of the strongly connected components of D).

Observe that since C_1, \dots, C_p are pairwise disjoint, we have

$$\text{OLA}(D) \geq \text{OLA}(D[C_1]) + \dots + \text{OLA}(D[C_p]).$$

On the other hand, there exists a vertex ordering of D of cost equal to $\text{OLA}(D[C_1]) + \dots + \text{OLA}(D[C_p])$: one may simply concatenate optimum-cost orderings of $D[C_1], D[C_2], \dots, D[C_p]$. This means that in fact

$$\text{OLA}(D) = \text{OLA}(D[C_1]) + \dots + \text{OLA}(D[C_p]). \quad (1)$$

Now observe that if any strongly connected component C_i has only one vertex, then its removal from the digraph does not change the OLA-cost. Let then D' be obtained from D by removing

all the strongly connected components of size one; then $\text{OLA}(D') = \text{OLA}(D)$. Clearly, if D' has at most $2k$ vertices, then it can be reported by the algorithm. We argue that otherwise, if D' has more than $2k$ vertices, then the algorithm can conclude that $\text{OLA}(D) = \text{OLA}(D') > k$.

Take any connected component C_i , and examine $D[C_i]$. Since $D[C_i]$ is strongly connected, for any its vertex ordering π , all the cuts $E(\pi_{>j}, \pi_{\leq j})$ for $j \in \{1, 2, \dots, |C_i| - 1\}$ will be nonempty. Consequently, the cost of any ordering π will be at least $|C_i| - 1$, so

$$\text{OLA}(C_i) \geq |C_i| - 1. \quad (2)$$

Observe that if $|C_i| > 1$ then $|C_i| - 1 \geq |C_i|/2$. Hence, by (1) and (2) we have

$$\text{OLA}(D) = \sum_{i=1}^p \text{OLA}(D[C_i]) \geq \sum_{i=1}^p (|C_i| - 1) \geq \frac{1}{2} \sum_{i: |C_i| > 1} |C_i| = \frac{1}{2} |V(D')|.$$

We conclude that if D' has more than $2k$ vertices, then $\text{OLA}(D) > k$. \square

Let us now turn to the CUTWIDTH problem in semi-complete digraphs. We show that this problem admits a simple AND-composition.

Theorem 5. *Unless $\text{NP} \subseteq \text{coNP}/\text{poly}$, there exists no polynomial-size kernelization algorithm for CUTWIDTH problem in semi-complete digraphs when parameterized by the target width.*

Proof. We give an AND-composition algorithm for the problem. Let $(D_1, c), (D_2, c), \dots, (D_t, c)$ be the input instances, where each digraph D_i is semi-complete. The algorithm constructs a semi-complete digraph D^* by taking the disjoint union of D_1, D_2, \dots, D_t , and for all $1 \leq i < j \leq t$, adding arcs (u, v) for all $u \in V(D_i)$ and $v \in V(D_j)$. The output instance is (D^*, c) , so it remains to argue that $\text{ctw}(D^*) \leq c$ if and only if $\text{ctw}(D_i) \leq c$ for all $i = 1, 2, \dots, t$. In one direction, if each D_i admits a vertex ordering π_i of width at most c , then the concatenation $\pi_1 \cdot \pi_2 \cdot \dots \cdot \pi_t$ of these orderings is a vertex ordering of D^* of width at most c . Conversely, each D_i is an induced subdigraph of D^* , so if $\text{ctw}(D_i) > c$ then also $\text{ctw}(D^*) > c$. We conclude that indeed we have $\text{ctw}(D^*) \leq c$ if and only if $\text{ctw}(D_i) \leq c$ for all $i = 1, 2, \dots, t$.

Now the statement directly follows from Theorem 1 and the fact that computing the cutwidth of a semi-complete digraph is an NP-complete problem (see Theorem 14 in Section 6). \square

4.2 A Turing kernel for the CUTWIDTH problem

We now prove that an instance of the CUTWIDTH problem can be reduced to polynomially many instances of quadratic size which altogether are equivalent to the original instance; this yields a quadratic Turing kernel for the problem, as announced in Section 1. The following technical statement, which we will prove next, describes the output of this procedure in details.

Proposition 6. *There exists a polynomial-time algorithm that given a semi-complete digraph D and integer c , either correctly concludes that $\text{ctw}(D) > c$ or outputs a list of at most n induced subdigraphs D_1, \dots, D_ℓ of D , each with at most $24c^2 + 40c + 1$ vertices, such that $\text{ctw}(D) \leq c$ if and only if $\text{ctw}(D_i) \leq c$ for each $i \in \{1, 2, \dots, \ell\}$.*

Proposition 6 immediately implies that the problem admits a Turing kernel.

Theorem 7. *The CUTWIDTH problem in semi-complete digraphs, parameterized by the target width c , admits a Turing kernel with $\mathcal{O}(n)$ calls to an oracle solving instances of size $\mathcal{O}(c^2)$.*

Proof. Apply the algorithm of Proposition 6 and use an oracle call for each output induced subdigraph D_i to resolve whether its cutwidth is at most c . \square

The remainder of this section is devoted to the proof of Proposition 6. The line of the reasoning is as follows. We first compute a 2-approximate ordering using Theorem 3, and then we exhaustively improve it until it becomes a lean ordering (see definition below). Let σ be the obtained ordering, and consider the sequence of cutsizes along σ . The next observation is crucial. Due to leanness, if some cutsize in this sequence is smaller or equal than $\Omega(c)$ cutsizes to the left and to the right, then there is some optimum-width ordering that uses the corresponding cut; that is, the prefix of σ up to this cut is also a prefix of some optimum-width ordering. We call such cuts *milestones*. It is not hard to prove that a milestone can be found every $\mathcal{O}(c^2)$ vertices in the ordering σ . Thus we are able to partition the digraph into pieces of size $\mathcal{O}(c^2)$ that may be treated independently. Each of these pieces gives rise to one digraph D_i in the output of the kernelization algorithm.

Before proceeding to the proof itself, we first prove an auxiliary statement that intuitively says the following: if two vertices are far from each another in a given ordering of width at most c , then their relative ordering remains unchanged in any other ordering of width at most c .

Lemma 2. *Let $D = (V, E)$ be a semi-complete digraph and let π be its vertex ordering of width at most c . Suppose $u, v \in V$ are such that $\pi(u) < \pi(v) - 3c$. Then, for every vertex ordering σ of D of width at most c , we have that u is placed before v in σ .*

Proof. Let X be the set of vertices placed between u and v in π (exclusive). As $\pi(u) < \pi(v) - 3c$, we have $|X| \geq 3c$. We are going to exhibit $c + 1$ arc-disjoint paths leading from u to v . This implies that in any ordering σ of width at most c , u must be placed before v , as otherwise each of these paths would contribute to the size of every cut in σ between v and u .

Suppose first that $(v, u) \in E$. Observe that $|X \cap N^-(u)| \leq c - 1$, since each of arc with tail in $X \cap N^-(u)$ and head being u contributes to the size of the cut $E_\pi^{\pi(u)}$, and we moreover have the arc $(v, u) \in E$. Similarly, $|X \cap N^+(v)| \leq c - 1$. If we now denote $Y = X \setminus (N^-(u) \cup N^+(v))$, then we infer that $|Y| \geq 3c - 2(c - 1) = c + 2$. Since D is semi-complete, we have that for each $w \in Y$ there are arcs (u, w) and (w, v) , which form a path of length 2 from u to v . This is a family of $c + 2$ arc-disjoint paths from u to v .

Suppose now that $(v, u) \notin E$, hence $(u, v) \in E$ since D is semi-complete. Exactly as above we infer that $|X \cap N^-(u)| \leq c$ and $|X \cap N^+(v)| \leq c$ (we now do not have the additional arc (v, u)), which implies that $|Y| \geq 3c - 2c = c$. Again, vertices of Y give rise to a family of c arc-disjoint paths of length 2 from u to v , to which we can add the one-arc path consisting of (u, v) to get $c + 1$ arc-disjoint paths from u to v . \square

The essence of our approach is encapsulated in Lemma 3 below, which provides a sufficient condition for a cut in a given ordering π so that it can be assumed to be used in an optimum ordering σ .

Lemma 3. Let $D = (V, E)$ be a semi-complete digraph. Let π and σ be two vertex orderings of D such that $\text{ctw}(D, \sigma) \leq \text{ctw}(D, \pi) = c$. Suppose further that $m \in [3c, |V| - 3c]$ is such that in D there is a family of $|E_\pi^m|$ arc-disjoint paths leading from $\pi_{>m+3c}$ to $\pi_{\leq m-3c}$. Then there exists a vertex ordering σ^* such that:

- $\sigma_{\leq m}^* = \pi_{\leq m}$;
- for every j with $j \leq m - 3c$ or $j > m + 3c$, we have $\sigma_j^* = \sigma_j$;
- $\text{ctw}(D, \sigma^*) \leq \text{ctw}(D, \sigma)$.

The intuition behind Lemma 3 is as follows. Consider σ^* as rearranged σ . The second condition says that the rearrangement is local: it affects only vertices at positions in the range $[m-3c+1, m+3c]$. The third condition says that the rearrangement does not increase the width. Finally, the first condition is crucial: σ^* uses the prefix $\pi_{\leq m}$ of π as one of its prefixes. Thus, the intuition is that any ordering can be locally rearranged while preserving the width so that prefix $\pi_{\leq m}$ is used, provided there is a large arc-disjoint flow locally near m . See Figure 2 below.

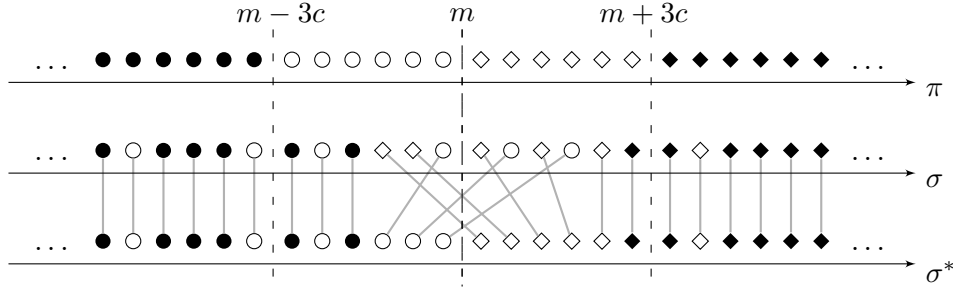


Fig. 2: Situation in the proof of Lemma 3. Vertex orderings π , σ , and σ^* are depicted, where equal vertices in σ and σ^* are connected by a grey line. Circle nodes belong to $\pi_{\leq m}$, and diamond nodes belong to $\pi_{>m}$. White nodes belong to $\pi_{>m-3c} \cap \pi_{\leq m+3c}$. Claim 1 says that the black nodes cannot be swapped in σ with a node of the complementary shape.

Proof of Lemma 3. We first establish the following basic observation on the relation between orderings π and σ .

Claim 1. In the ordering σ , every vertex of $\pi_{\leq m-3c}$ is placed before every vertex of $\pi_{>m}$, and every vertex of $\pi_{\leq m}$ is placed before every vertex of $\pi_{>m+3c}$.

Proof. Consider any $u \in \pi_{\leq m-3c}$ and $v \in \pi_{>m}$. Then, $\pi(u) < \pi(v) - 3c$, so the first statement follows immediately from Lemma 2. The proof of the second statement is the same. \square

Let σ_{\leq} and $\sigma_{>}$ denote the restriction of σ to $\pi_{\leq m}$ and $\pi_{>m}$, respectively. Then, define σ^* to be the concatenation of σ_{\leq} and $\sigma_{>}$; see Figure 2 for reference. By the construction we have $\pi_{\leq m} = \sigma_{\leq}^*$, so the first condition is satisfied. For the second condition, observe that by Claim 1, every vertex of $\pi_{\leq m-3c}$ is before every vertex of $\pi_{>m}$ in σ . It follows that in σ , the first vertex of $\pi_{>m}$ appears only after a prefix of at least $m - 3c$ vertices of $\pi_{\leq m}$. In the construction of σ^* from σ , the vertices of that prefix stay at their original positions, so $\sigma_j^* = \sigma_j$ for all $j \leq m - 3c$. A symmetric argument shows that $\sigma_j^* = \sigma_j$ also for all $j > m + 3c$.

It remains to prove that $\text{ctw}(D, \sigma^*) \leq \text{ctw}(D, \sigma)$. Consider any $j \in [0, |V|]$; we need to prove that $|E_{\sigma^*}^j| \leq \text{ctw}(D, \sigma)$. By the second condition we have that $E_{\sigma^*}^j = E_{\sigma}^j$ when $j \leq m - 3c$ or $j \geq m + 3c$, and $|E_{\sigma}^j| \leq \text{ctw}(D, \sigma)$ by definition. Hence, we are left with checking the inequality for j satisfying $m - 3c < j < m + 3c$.

In the following, for a vertex subset A we denote $\delta(A) = |E(V \setminus A, A)|$. We will use the submodularity of directed cuts: $\delta(A \cap B) + \delta(A \cup B) \leq \delta(A) + \delta(B)$ for all vertex subsets A, B . In these terms, we need to prove that $\delta(\sigma_{\leq j}^*) \leq \text{ctw}(D, \sigma)$.

Let x be the vertex at position j in σ^* and let X be the set containing x and all vertices placed before x in σ . Suppose first that $j \leq m$. Then, by the construction we have $x \in \pi_{\leq m}$ and $\sigma_{\leq j}^* = X \cap \pi_{\leq m}$. By the submodularity of cuts we have

$$\delta(\sigma_{\leq j}^*) = \delta(X \cap \pi_{\leq m}) \leq \delta(X) + \delta(\pi_{\leq m}) - \delta(X \cup \pi_{\leq m}). \quad (3)$$

As X is a prefix of σ by definition, we have $\delta(X) \leq \text{ctw}(D, \sigma)$. Hence, by (3), in order to prove that $\delta(\sigma_{\leq j}^*) \leq \text{ctw}(D, \sigma)$, it suffices to prove that $\delta(X \cup \pi_{\leq m}) \geq \delta(\pi_{\leq m})$.

Denote $d = \delta(\pi_{\leq m}) = |E_{\pi}^m|$ and recall that there is a family of d arc-disjoint paths leading from $\pi_{>m+3c}$ to $\pi_{\leq m-3c}$. In particular, this means that for each set A with $A \supseteq \pi_{\leq m-3c}$ and $A \cap \pi_{>m+3c} = \emptyset$, each of these paths has to contribute to $\delta(A)$, implying $\delta(A) \geq d$.

Therefore, it suffices to show that $X \cup \pi_{\leq m} \supseteq \pi_{\leq m-3c}$ and $(X \cup \pi_{\leq m}) \cap \pi_{>m+3c} = \emptyset$. While the first assertion is trivial, the second is equivalent to $X \cap \pi_{>m+3c} = \emptyset$. For this, observe that by definition no vertex of X is placed after x in σ , and x belongs to $\pi_{\leq m}$. Moreover, by Claim 1 all vertices of $\pi_{>m+3c}$ are placed in σ after all vertices of $\pi_{\leq m}$, in particular after x . This implies that X and $\pi_{>m+3c}$ are disjoint. By the discussion above, this proves that $\delta(X \cup \pi_{\leq m}) \geq d$ and, consequently also that $\delta(\sigma_{\leq j}^*) \leq \text{ctw}(D, \sigma)$.

The proof for the case $j > m$ is completely symmetric, however we need to observe that now $x \in \pi_{>m}$ and $\sigma_{\leq j}^* = X \cup \pi_{\leq m}$. By applying the same submodularity argument (3), we are left with proving that $\delta(X \cap \pi_{\leq m}) \geq \delta(\pi_{\leq m})$, which follows by a symmetric reasoning. \square

Our goal now is to construct an approximate ordering π where we will be able to find many positions m to which Lemma 3 can be applied. We first recall the concept of a lean ordering, which will be our main tool for finding families of arc-disjoint paths.

Definition 1. A vertex ordering π of a digraph $D = (V, E)$ is called *lean* if for each $0 \leq a \leq b \leq n$, the maximum size of a family of arc-disjoint paths from $\pi_{>b}$ to $\pi_{\leq a}$ in D is equal to $\min_{a \leq i \leq b} |E_{\pi}^i|$.

Note that by Menger's theorem, the maximum size of a family of arc-disjoint paths from $\pi_{>b}$ to $\pi_{\leq a}$ is equal to the minimum size of a cut separating $\pi_{>b}$ from $\pi_{\leq a}$. Thus, in a lean ordering we have that the minimum cutsize between any disjoint prefix and suffix is actually realized by one of the cuts along the ordering.

The notion of a lean ordering is the cutwidth analogue of a *lean decomposition* in the treewidth setting, cf. [24]. An essentially equivalent notion of *linked orderings* was used by Chudnovsky and Seymour [6] in the context of immersions in tournaments. Also, Giannopoulou et al. [16] used this concept to study immersion obstructions for the cutwidth of undirected graphs. A careful analysis of the arguments of [6, 16] yields the following.

Lemma 4 ([6, 16]). *There is a polynomial-time algorithm that given a vertex ordering π of a digraph D , computes a lean vertex ordering π^* of D satisfying $\text{ctw}(D, \pi^*) \leq \text{ctw}(D, \pi)$.*

Proof. To prove the statement we may adapt the proof of Lemma 3.1 from [6], or the proof of Lemma 13 from [16].

Suppose there exist two integers a and b certifying that π is not a lean ordering. That is, if we denote $d = \min_{i \in [a, b]} |E_\pi^i|$, then there is no family of d arc-disjoint paths leading from $\pi_{>b}$ to $\pi_{\leq a}$. By Menger's Theorem there exists a partition (A, B) of $V(D)$ such that $\pi_{\leq a} \subseteq A$, $\pi_{>b} \subseteq B$, and $|E(B, A)| < d$. Such a partition can be identified in polynomial time using a min-cut/max-flow algorithm.

Consider the vertex ordering π' obtained by concatenating the restriction π_A of π to A with the restriction of π_B of π to B ; that is, $\pi' = \pi_A \cdot \pi_B$. The following now holds:

Claim 2. *For every $i' \in [0, |V|]$, if the vertex $\pi'_{i'}$ is at position i in π , then we have that $|E_{\pi'}^{i'}| \leq |E_\pi^i|$, and moreover this inequality is strict for $i' = |A|$.*

The proof of Claim 2 follows easily from submodularity of cuts, see the proof of 3.1 in [6] and the proof of Lemma 13 in [16]; this latter paper considers undirected setting, but the submodularity argument holds in the directed setting in the same way. From Claim 2 it follows that $\text{ctw}(D, \pi') \leq \text{ctw}(D, \pi)$ and $\text{OLA}(D, \pi') < \text{OLA}(D, \pi)$ (note that Lemma 13 in [16] uses exactly this potential as the minimization goal).

Therefore, the algorithm starts with the input ordering π and as long as the leanness condition is not satisfied for some a and b , it computes the refined ordering π' as above and sets $\pi := \pi'$. Note that the width of the ordering cannot increase in this process, while the OLA-cost strictly decreases at each iteration. Since the OLA-cost is at most $|V|^3$ at the beginning, we infer that the algorithm outputs some lean ordering after at most a cubic number of iterations. \square

Next, we introduce the concept of a *milestone*. Intuitively, a milestone is a position where Lemma 3 can be applied, provided the ordering is lean.

Definition 2. *Let π be a vertex ordering of a digraph $D = (V, E)$, and let α be a positive integer. An integer $m \in [0, |V|]$ is a π -milestone of D of span α if $|E_\pi^m| \leq |E_\pi^i|$ for each integer i with $m - \alpha \leq i \leq m + \alpha$.*

Note that if π is lean and m is a π -milestone of span α , then $\min_{m-\alpha \leq i \leq m+\alpha} |E_\pi^i| = |E_\pi^m|$, hence there is a family of $|E_\pi^m|$ arc-disjoint paths leading from $\pi_{>m+\alpha}$ to $\pi_{\leq m-\alpha}$. Thus, a π -milestone of span $3c$ satisfies the prerequisite of Lemma 3 about the existence of arc-disjoint paths. We now observe that, in an ordering of small width, every large enough set of consecutive positions contains a milestone.

Lemma 5. *Let $D = (V, E)$ be a digraph and let π be a vertex ordering of D of width at most c . Then for any integers $p \in [0, |V|]$ and $\alpha \geq 0$, there exists a π -milestone $m \in [p - \alpha \cdot c, p + \alpha \cdot c]$ of span α .*

Proof. We look for a milestone by means of an iterative procedure. Initially set $m := p$. While m is not a π -milestone of span α , there exists an integer $j \in [m - \alpha, m + \alpha]$ such that $|E_\pi^j| < |E_\pi^m|$. Then continue the iteration setting $m := j$. Observe that with each iteration, the cutsize $|E_\pi^m|$

strictly decreases. Since $\text{ctw}(D, \pi) \leq c$, it follows that the number of the number of iterations before finding a π -milestone of span α is at most c . Each iteration replaces m with a number differing from it by at most α , hence the final π -milestone m satisfies $|p - m| \leq \alpha \cdot c$, as required. \square

With all the tools gathered, we can finish the proof of Proposition 6.

Proof of Proposition 6. By Theorem 3, we can compute in polynomial time a vertex ordering π_0 of D such that $\text{ctw}(D, \pi_0) \leq 2 \cdot \text{ctw}(D)$. If $\text{ctw}(D, \pi_0) > 2c$, we can conclude that $\text{ctw}(D) > c$ and report this answer, so let us assume that $\text{ctw}(D, \pi_0) \leq 2c$. By applying the algorithm of Lemma 4 to π_0 , we can compute in polynomial time a lean ordering π such that $\text{ctw}(D, \pi) \leq \text{ctw}(D, \pi_0) \leq 2c$. In the following we assume w.l.o.g. that $|V| > 12c$, for otherwise we can output a list consisting only of D .

Call a set of π -milestones *dispersed* if these π -milestone pairwise differ by more than $12c$. Observe that 0 and $|V|$ are always π -milestones, and they differ by more than $12c$. Starting from the set $\{0, |V|\}$, we compute an inclusion-wise maximal dispersed set $0 = m_0 < m_1 < m_2 < \dots < m_\ell = |V|$ of π -milestones of span $6c$. More precisely, whenever some π -milestone of span $6c$ can be added to the set without spoiling the dispersity requirement, we do it, until no further such milestone can be added. Observe that then we have that $m_{i+1} - m_i \leq 24c^2 + 24c + 1$ for each $i \in [1, \ell - 1]$, for otherwise the range $[m_i + 12c + 1, m_{i+1} - 12c - 1]$ would contain more than $24c^2$ vertices, so by Lemma 5 we would be able to find in it a π -milestone of span $6c$ that could be added to the constructed dispersed set.

Thus, π is partitioned into ℓ blocks B_1, \dots, B_ℓ , each of length at most $24c^2 + 24c + 1$, such that the j -th block B_j is equal to $\{\pi_{m_{j-1}+1}, \pi_{m_{j-1}+2}, \dots, \pi_{m_j}\}$. For each $j \in [1, \ell]$, let A_j be defined as B_j augmented with the following vertices:

- vertices at positions in ranges $[\max(1, m_{j-1} - 6c + 1), m_{j-1}]$ and $[m_j + 1, \min(|V|, m_j + 6c)]$,
- all heads of arcs from $E_\pi^{m_{j-1}-6c}$, and all tails of arcs from $E_\pi^{m_j+6c}$.

Since the width of π is at most $2c$, we have that $|A_j| \leq |B_j| + 16c \leq 24c^2 + 40c + 1$.

For $j \in [1, \ell]$, let us denote $D_j = D[A_j]$. To prove the theorem, it now suffices to show that $\text{ctw}(D) \leq c$ if and only if $\text{ctw}(D_j) \leq c$ for each $j \in [1, \ell]$. The forward direction is trivial, since cutwidth is closed under taking induced subdigraphs. Hence, we are left with showing that if $\text{ctw}(D_j) \leq c$ for each $j \in [1, \ell]$, then $\text{ctw}(D) \leq c$.

Take any $j \in [1, \ell - 1]$. As m_j is a π -milestone of span $6c$, we have $\min_{i \in [m_j-6c, m_j+6c]} |E_\pi^i| = |E_\pi^{m_j}|$.

Since π is lean, there is a family \mathcal{F}_j of $|E_\pi^{m_j}|$ arc-disjoint paths in D leading from $\pi_{>m_j+6c}$ to $\pi_{\leq m_j-6c}$. We can assume w.l.o.g. that each internal (non-endpoint) vertex of each of these paths has position between $m_j + 6c$ and $m_j - 6c + 1$ in π . Hence, in particular, each path of \mathcal{F}_j starts with an arc of $E_\pi^{m_j+6c}$ and ends with an arc of $E_\pi^{m_j-6c}$. This implies that for each $j \in [1, \ell]$, all the paths of \mathcal{F}_j are entirely contained both in D_j and in D_{j+1} .

Consider any $j \in [1, \ell]$, and for simplicity assume for now that $j \neq 1$ and $j \neq \ell$. Let π' be the restriction of π to the vertex set of D_j ; obviously the width of π' is at most $2c$. Further, let m' be the position of $\pi_{m_{j-1}}$ in π' , so that $\pi'_{\leq m'} = \pi_{\leq m_{j-1}} \cap V(D_j)$. Observe that since all vertices at positions between $m_{j-1} - 6c + 1$ and $m_{j-1} + 6c$ in π are included in the vertex set of D_j , they are at positions between $m' - 6c + 1$ and $m' + 6c$ in π' , and hence the paths of \mathcal{F}_{j-1}

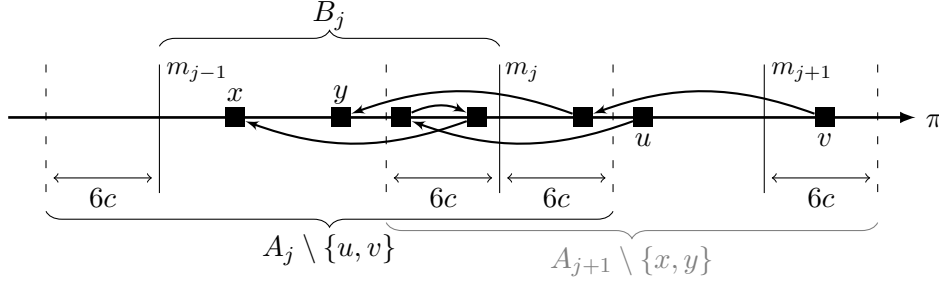


Fig. 3: The thick arcs represent the paths of \mathcal{F}_j that are contained both in D_j and in D_{j+1} .

in D_j lead from $\pi'_{>m'+6c}$ to $\pi'_{\leq m'-6c}$. Their number is $|E_{\pi}^{m_{j-1}}|$, which is equal to the cutsize at position m' in π' , by the construction of D_j and π' .

We conclude that Lemma 3 can be applied to position m' in the ordering π' of D_j . If we now use it on any ctw -optimal vertex ordering σ of D_j , we obtain a ctw -optimal vertex ordering σ^* of D_j such that $\sigma^*_{\leq m'} = \pi'_{\leq m'} = \pi_{\leq m_{j-1}} \cap V(D_j)$. Note that by Lemma 3, σ^* differs from σ by a rearrangement of vertices at positions between $m' - 6c + 1$ and $m' + 6c$.

Now we define m'' to be the position of π_{m_j} in π' , so that $\pi'_{\leq m''} = \pi_{\leq m_j} \cap V(D_j)$. A symmetric reasoning, which uses the fact that \mathcal{F}_j is also entirely contained in D_j , shows that Lemma 3 can be also applied to position m'' in the ordering π' of D_j . Then we can use this lemma on the ctw -optimal vertex ordering σ^* , yielding a ctw -optimal ordering σ^{**} such that $\sigma^{**}_{\leq m''} = \pi'_{\leq m''} = \pi_{\leq m_j} \cap V(D_j)$. Again, by Lemma 3 we have that σ^* and σ^{**} differ by a rearrangement of vertices at positions $m'' - 6c + 1$ and $m'' + 6c$. Since $m_j - m_{j-1} > 12c$ by construction, we infer that this rearrangement does not change the prefix of length m' , and hence we still have $\sigma^{**}_{\leq m'} = \pi'_{\leq m'} = \pi_{\leq m_{j-1}} \cap V(D_j)$. The ordering σ^{**} obtained in this manner shall be called σ^j . For $j = 1$ and $j = \ell$ we obtain σ^j in exactly the same way, except we apply Lemma 3 only once, for the position not placed at the end of the sequence.

All in all, for each $j \in [1, \ell]$ we have obtained a ctw -optimal ordering σ^j of D_j such that the vertices of B_j form an infix (a sequence of consecutive elements) of σ^j , while vertices to the left of this infix are the vertices of $V(D_j) \cap \pi_{\leq m_{j-1}}$ and vertices to the right of this infix are the vertices of $V(D_j) \cap \pi_{>m_j}$. Define an ordering $\tilde{\sigma}$ of D by first restricting every ordering σ^j to B_j , and then concatenating all the obtained orderings for $j = 1, 2, \dots, \ell$. Since we assumed that $\text{ctw}(D_j) \leq c$ for each $j \in [1, \ell]$, and each ordering σ^j is ctw -optimal on D_j , we have that $\text{ctw}(D_j, \sigma^j) \leq c$ for each $j \in [1, \ell]$. From the construction of D_j , and in particular the fact that all the arcs of $E_{\pi}^{m_{j-1}}$ and $E_{\pi}^{m_j}$ are contained in D_j , it follows that the infix of cutvector $\text{cuts}\langle D^j, \sigma^j \rangle$ corresponding to the vertices of B_j is equal to the infix of the cutvector $\text{cuts}\langle D, \tilde{\sigma} \rangle$ corresponding to the vertices of B_j . This shows that

$$\text{ctw}(D, \tilde{\sigma}) = \max_{i \in [0, |V|]} \text{cuts}\langle D, \tilde{\sigma} \rangle(i) \leq \max_{\substack{j \in [0, \ell] \\ i \in [0, |V(D_j)|]}} \text{cuts}\langle D^j, \sigma^j \rangle(i) = \max_{j \in [0, \ell]} \text{ctw}(D_j, \sigma^j) \leq c,$$

hence we are done. \square

5 Cutwidth-minimal semi-complete digraphs

Recall here that a digraph D is called c -cutwidth-minimal if the cutwidth of D is at least c , but the cutwidth of every proper induced subdigraph of D is smaller than c . In this section we provide upper bounds on the sizes of c -cutwidth-minimal tournaments and semi-complete digraphs. It turns out that the number of vertices in any c -cutwidth-minimal semi-complete digraph is bounded quadratically in c (see Theorem 9 below), while for c -cutwidth-minimal tournaments we can even give an almost tight upper bound that is linear in c (see Theorem 8 below). Essentially, the first result follows easily by considering applying the algorithm of Theorem 7 on a c -cutwidth-minimal semi-complete digraph for parameter $c-1$. For the second result we use the understanding of minimum orderings in tournaments in the spirit of Lemma 1. Finally, we also discuss direct algorithmic applications of both these theorems.

5.1 Upper bound for c -cutwidth-minimal tournaments

We first provide a linear bound on the sizes of c -cutwidth-minimal tournaments. Our main tool will be the notion of a *degree tangle*, introduced in [20] as a certificate for large cutwidth.

Definition 3. For a digraph D and nonnegative integers k and α , a (k, α) -degree tangle is a vertex set $W \subseteq V(D)$ such that $|W| \geq k$, and for each $u, v \in W$ we have $|d^-(u) - d^-(v)| \leq \alpha$.

In [20, 21] it is essentially shown that if a semi-complete digraph D admits a (k, α) -degree tangle, then the cutwidth of D is at least linear in $k - \alpha$. In (fractional) tournaments we can establish a quadratic bound, as shown next. Relying on the characterization of minimum orderings (Lemma 1), Lemma 6 below provides a slightly finer understanding of the relation between degree tangles and cutwidth than [20, 21].

Lemma 6. Let k and α be nonnegative integers. If a fractional tournament T contains a $(2k+1, \alpha)$ -degree tangle, then $\text{ctw}(T) \geq \frac{k(k+1-\alpha)}{2}$.

Proof. Let W be the $(2k+1, \alpha)$ -degree tangle present in the fractional tournament T . Consider a sorted ordering π of T . By Lemma 1, we have $0 \leq \text{cuts}\langle T, \pi \rangle(i) \leq \text{ctw}(T)$ for any $i \in [0, n]$. Observe that

$$\text{cuts}\langle T, \pi \rangle(i) = \omega(V, \pi_{\leq i}) - \omega(\pi_{\leq i}, \pi_{\leq i}) = \sum_{u \in \pi_{\leq i}} \omega^-(u) - \binom{i}{2} = \sum_{j=1}^i (\omega^-(\pi_j) - (j-1)).$$

We infer that for any position $p \in [0, n-2k]$, the two following inequalities hold:

$$\sum_{j=p}^{p+k-1} (\omega^-(\pi_j) + 1 - j) = \text{cuts}\langle T, \pi \rangle(p+k-1) - \text{cuts}\langle T, \pi \rangle(p-1) \leq \text{ctw}(T) \quad (4)$$

$$\sum_{j=p+k+1}^{p+2k} -(\omega^-(\pi_j) + 1 - j) = \text{cuts}\langle T, \pi \rangle(p+k) - \text{cuts}\langle T, \pi \rangle(p+2k) \leq \text{ctw}(T) \quad (5)$$

Set $p = \min_{v \in W} \pi(v)$, that is, p is the lowest position occupied by a vertex of W . Let $\delta = \omega^-(\pi_p)$. Since π is sorted, without loss of generality we may assume that $W = \{\pi_j : j \in [p, p+2k]\}$. As W is a $(2k+1, \alpha)$ -degree tangle, for each $j \in [p, p+2k]$, we have $\delta \leq \omega^-(\pi_j) \leq \delta + \alpha$.

Suppose $\delta + 1 \geq p + k - \frac{\alpha}{2}$. We use (4) together with $\omega^-(\pi_j) \geq \delta$ and re-indexing $i' = p + k - j$ to get:

$$\frac{k(k+1-\alpha)}{2} = \sum_{i'=1}^k (i' - \frac{\alpha}{2}) = \sum_{j=p}^{p+k-1} (p+k-j - \frac{\alpha}{2}) \leq \sum_{j=p}^{p+k-1} (\delta + 1 - j) \leq \text{ctw}(T).$$

Otherwise, we have $\delta + 1 < p + k - \frac{\alpha}{2}$. Then we use (5) together with $\omega^-(\pi_j) \leq \delta + \alpha$ and re-indexing $i' = j - p - k$ to get:

$$\frac{k(k+1-\alpha)}{2} = \sum_{i'=1}^k (i' - \frac{\alpha}{2}) = \sum_{j=p+k+1}^{p+2k} (j - p - k - \frac{\alpha}{2}) < \sum_{j=p+k+1}^{p+2k} -(\delta + \alpha + 1 - j) \leq \text{ctw}(T).$$

In both cases, we conclude that $\frac{k(k+1-\alpha)}{2} \leq \text{ctw}(T)$. \square

We are now ready to show that c -cutwidth-minimal tournaments have sizes linear in c .

Theorem 8. *For every $c \in \mathbb{N}$, every c -cutwidth-minimal tournament has at most $2c + 2\lceil\sqrt{2c}\rceil + 1$ vertices.*

Proof. Consider a c -cutwidth-minimal tournament $T = (V, E)$ and let π be a sorted ordering of T . By Lemma 1, we have $\text{ctw}(T, \pi) = \text{ctw}(T) \geq c$. Let $i \in [0, |V|]$ be such that $\text{cuts}(T, \pi)(i) \geq c$. We define the vertex set $V_{=} = \{v \in V : d^-(v) = d^-(\pi_i)\}$. We let $V_{>}$ and $V_{<}$ denote the sets of vertices that respectively appear after and before $V_{=}$ in π .

Observe that for each $u \in V_{=}$, the set $V_{=} \setminus \{u\}$ is a $(|V_{=}| - 1, 1)$ -degree tangle for $T[V \setminus \{u\}]$. By the minimality of T and Lemma 6, we may thus assume that $|V_{=}| \leq 2\lceil\sqrt{2c}\rceil + 1$, as otherwise the removal of any vertex of $V_{=}$ would still leave a degree tangle that certifies that the cutwidth is at least c .

We now focus on the set $V_{>}$. Note that $V_{<} \cup V_{=}$ is a prefix of π and $V_{>}$ is a suffix of π . Consider now removing any vertex v of $V_{>}$ from the tournament T . This operation may decrease the indegrees of vertices of $V_{>}$ by at most one, so after the removal it will be still true that the indegree of any vertex of $V_{>}$ will be at least as large as the indegree of any vertex of $V_{<} \cup V_{=}$. Consequently, there is a sorted vertex ordering of $T[V \setminus \{v\}]$ where $V_{<} \cup V_{=}$ is a prefix and $V_{>} \setminus \{v\}$ is a suffix. Moreover, if v had no outneighbors in $V_{<} \cup V_{=}$, then we could choose this sorted ordering so that on $V_{<} \cup V_{=}$ it would match π , implying that E_{π}^i would be also a cut; note here that no arc of E_{π}^i is incident to v , as v has no outneighbors in $V_{<} \cup V_{=}$. By Lemma 1, this would mean that the cutwidth of $T[V \setminus \{v\}]$ would be at least $|E_{\pi}^i|$, which is at least c , a contradiction to the minimality of T . We conclude that each $v \in V_{>}$ has an outneighbor in $V_{<} \cup V_{=}$.

Since there is a sorted ordering of $T[V \setminus \{v\}]$ where $V_{<} \cup V_{=}$ is a prefix and $V_{>} \setminus \{v\}$ is a suffix, by Lemma 1 we infer that the cutwidth of $T[V \setminus \{v\}]$ has to be at least $|E(V_{>} \setminus \{v\}, V_{<} \cup V_{=})|$. However, we have just argued that every vertex of $V_{>}$ has an outneighbor in $V_{<} \cup V_{=}$, so $|E(V_{>} \setminus \{v\}, V_{<} \cup V_{=})| \geq |V_{>} \setminus \{v\}|$. By the minimality of T , this implies that $|V_{>} \setminus \{v\}| \leq \text{ctw}(T[V \setminus \{v\}]) < c$, so $|V_{>}| \leq c$. A symmetric argument shows that $|V_{<}| \leq c$ as well. The claim follows from combining the obtained upper bounds on the sizes of $V_{<}$, $V_{=}$, and $V_{>}$. \square

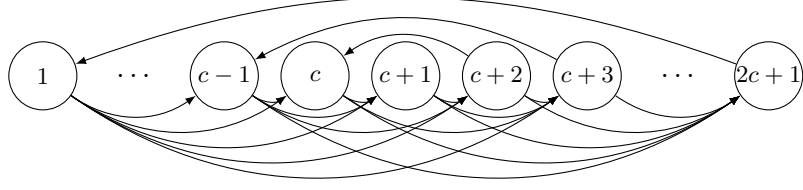


Fig. 4: A c -cutwidth-minimal tournament T with $2c+1$ vertices. In the layout π depicted above, the set of backward arcs of T forms the matching $\{(\pi_{c+1+i}, \pi_{c+1-i}) : i \in [1, c]\}$.

Note that the bound of Theorem 8 is almost tight, as Figure 4 displays a c -cutwidth-minimal tournament with $2c+1$ vertices; we leave the easy verification of minimality to the reader. We also remark that the example in Figure 4 may be modified by replacing the depicted matching of backwards arcs by any matching of backwards arcs of size c with tails at positions between $c+2$ and $2c+1$ and heads at positions between 1 and c . This yields an exponential number of pairwise non-isomorphic c -cutwidth-minimal tournaments on $2c+1$ vertices.

5.2 Upper bound for c -cutwidth-minimal semi-complete digraphs

We now prove the quadratic bound on the sizes of c -cutwidth-minimal semi-complete digraphs.

Theorem 9. *For every positive integer c , every c -cutwidth-minimal semi-complete digraph has at most $24c^2 + 1$ vertices.*

Proof. For $c = 0$ and $c = 1$ the theorem holds trivially, since the only 0-cutwidth-minimal semi-complete digraph is an isolated vertex and the only two 1-cutwidth-minimal semi-complete digraphs are the following: two vertices connected by a pair of symmetric arcs, and an oriented triangle (directed cycle on 3 vertices). Thus we may assume that $c > 1$. Let D be a c -cutwidth-minimal semi-complete digraph. We may assume that D is not a tournament, as otherwise Theorem 8 applies proving the bound.

We say that a c -cutwidth-minimal semi-complete digraph D is *c -critical* if every semi-complete subdigraph D' of D resulting from the removal of an arc satisfies $\text{ctw}(D') < c$. We claim that it is enough to prove the statement for c -critical semi-complete digraphs. Indeed, observe that every c -critical semi-complete subdigraph D' of D satisfies $V(D) = V(D')$, because otherwise, by the minimality of D , we would have $c \leq \text{ctw}(D') \leq \text{ctw}(D[V(D')]) < c$, a contradiction. So assume that D is a c -critical semi-complete digraph. We now claim that $\text{ctw}(D) = c$. To see this consider any semi-complete digraph D' obtained by removing an arc e . Since D is not a tournament, such D' exists and, by c -criticality, it satisfies $c \leq \text{ctw}(D) \leq \text{ctw}(D') + 1 \leq c$. This implies $\text{ctw}(D) = c$.

Consider applying the algorithm of Proposition 6 to the semi-complete digraph D with parameter $c-1$. Observe that the first step of this algorithm is to compute a vertex ordering π of D of width at most $2c-2$ using the approximation algorithm of Theorem 3. In case the application of Theorem 3 does not return such an ordering, the algorithm terminates and concludes that $\text{ctw}(D) > c-1$. Moreover, this conclusion may be drawn only in such case: if the algorithm of Theorem 3 succeeds in finding a vertex ordering π of width at most $2c-2$,

the algorithm of Proposition 6 returns a list of induced subdigraphs D_1, \dots, D_ℓ satisfying the asserted properties. However, since $\text{ctw}(D) = c$ and $c \leq 2c - 2$ due to $c \geq 2$, we may apply the algorithm of Proposition 6 on any ordering π of D of width c , instead of the one obtained by applying algorithm of Proposition 6. Thus we guarantee that the algorithm always produces a list of induced subdigraphs D_1, \dots, D_ℓ with the asserted properties: for every $i \in [1, \ell]$, D_i is an induced subdigraph of D on at most $24(c-1)^2 + 40(c-1) + 1 \leq 24c^2 + 1$ vertices, and $\text{ctw}(D) \leq c - 1$ if and only if $\text{ctw}(D_i) \leq c - 1$ for each $i \in [1, \ell]$.

If we now had that each output D_i was smaller by at least one vertex than D , then by the c -cutwidth minimality of D we would infer that $\text{ctw}(D_i) \leq c - 1$ for each $i \in [1, \ell]$, implying $\text{ctw}(D) \leq c - 1$. This is a contradiction with the assumption $\text{ctw}(D) = c$. Hence, for some $i \in [1, \ell]$ we have $D_i = D$, and hence $|V(D)| \leq 24c^2 + 1$. \square

Observe that in the previous proof, to apply the algorithm of Proposition 6, we use the fact that $\text{ctw}(D) = c \leq 2c - 2$ and then consider an ordering π of optimal width c instead of a 2-approximation. Analyzing in details the algorithm of Proposition 6 in this context would lead to smaller constants in the bound on the size of c -cutwidth minimal semi-complete digraphs, namely $6c^2 + O(c)$.

5.3 Algorithmic applications

Consider the c -CUTWIDTH VERTEX DELETION problem defined as follows: given a digraph D and integer k , decide whether it is possible to find a set Z of at most k vertices in D such that $\text{ctw}(D - Z) \leq c$; here, c is considered a fixed constant. A set Z with this property is called a *deletion set to cutwidth at most c* .

The upper bound on the sizes of c -cutwidth-minimal semi-complete digraphs, proved in the previous section, yields almost directly a number of algorithmic corollaries for the c -CUTWIDTH VERTEX DELETION problem. More precisely, we show that for a fixed c , we can give

- a single-exponential parameterized algorithm, with the running time tight under ETH;
- an approximation algorithm with the approximation factor depending only on c ; and
- a polynomial kernelization algorithm.

As a preparation for these results, we first prove that it is possible to efficiently locate small obstacles for cutwidth at most c in a semi-complete digraph of cutwidth larger than c .

Lemma 7. *There exists an algorithm that given a semi-complete digraph D on n vertices and a nonnegative integer c , runs in time $2^{O(\sqrt{c \log c})} \cdot n^3$ and either concludes that $\text{ctw}(D) \leq c$, or finds an induced $(c + 1)$ -cutwidth-minimal subdigraph of D .*

Proof. Recall that Fomin and Pilipczuk [11, 21] gave an algorithm that verifies whether the cutwidth of a given n -vertex semi-complete digraph is at most c in time $2^{O(\sqrt{c \log c})} \cdot n^2$. Given a semi-complete digraph D , we use this algorithm first to verify whether $\text{ctw}(D) \leq c$. If this is not the case, we perform the following procedure.

Set $D' := D$, iterate through the vertices of D , and for each consecutive vertex u check using the algorithm of Fomin and Pilipczuk whether $\text{ctw}(D' - u) \leq c$. If this is the case, then keep u in D' and proceed, and otherwise remove u from D' and proceed. Note that thus D' changes over the course of the algorithm but stays an induced subdigraph of D . We claim that

D'' , defined as D' obtained at the end of the procedure, is $(c + 1)$ -cutwidth-minimal. On one hand, we remove a vertex from D' only when this does not lead to decreasing the cutwidth below $c + 1$, so we maintain the invariant that the cutwidth of D' is always larger than c . On the other hand, each vertex u we keep in D' had the property that removing it would decrease the cutwidth to at most c at the moment it was considered. Since cutwidth is closed under induced subdigraphs, this is also true in D'' , so indeed D'' is $(c + 1)$ -cutwidth-minimal. The procedure applies the algorithm of Fomin and Pilipczuk $n + 1$ times, so the total running time is $2^{\mathcal{O}(\sqrt{c \log c})} \cdot n^3$. \square

We remark that in the tournament setting, computing the cutwidth is a polynomial-time solvable problem by Theorem 2. By plugging this subroutine instead of the algorithm of Fomin and Pilipczuk, we infer that for tournaments, the algorithm of Lemma 7 works in fully polynomial time, with no exponential multiplicative factor depending on c .

By applying a standard branching strategy, we obtain a single-exponential FPT algorithm for c -CUTWIDTH VERTEX DELETION in semi-complete digraphs.

Theorem 10. *The c -CUTWIDTH VERTEX DELETION problem on semi-complete digraphs can be solved in time $2^{\mathcal{O}(\sqrt{c \log c})} \cdot c^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$, where n is the number of vertices of the input digraph and k is the budget for the deletion set.*

Proof. Let D be the input digraph. Run the algorithm of Lemma 7 to either conclude that $\text{ctw}(D) \leq c$, or to find an induced $(c + 1)$ -cutwidth minimal semi-complete digraph P in D . By Theorem 9, P has at most $\mathcal{O}(c^2)$ vertices. Branch on which vertex of P is included in the solution; that is, for each vertex u of P recurse on the digraph $D - u$ with budget $k - 1$ for the deletion set. Whenever the budget drops to 0 and the considered digraph still does not have cutwidth at most c , we may discard the branch. Conversely, if the budget is nonnegative and the considered digraph has cutwidth at most k , then we have found a deletion set to cutwidth at most c of size at most k . The recursion tree has depth at most k and branching $\mathcal{O}(c^2)$, so the whole algorithm has running time $2^{\mathcal{O}(\sqrt{c \log c})} \cdot c^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. \square

Similarly, we obtain a constant-factor approximation algorithm by greedily removing each encountered $(c + 1)$ -cutwidth-minimal induced subdigraph.

Theorem 11. *There exists an algorithm that given a semi-complete digraph D on n vertices and a nonnegative integer c , runs in time $2^{\mathcal{O}(\sqrt{c \log c})} \cdot n^{\mathcal{O}(1)}$ and finds a deletion set to cutwidth at most c in D whose size is at most $\mathcal{O}(c^2)$ times larger than the minimum size of a deletion set to cutwidth at most c in D .*

Proof. Let D be the input digraph and let Y be a minimum-size deletion set to cutwidth at most c in D . Iteratively run the algorithm of Lemma 7 to either conclude that the current digraph, initially set to D , has cutwidth at most c , or to find an induced $(c + 1)$ -cutwidth minimal semi-complete digraph P in it. In the first case break the iteration and return the currently accumulated solution X , while in the second case remove all the vertices of P from the current digraph, include them in X (initially set to be empty), and continue the iteration. By the condition of breaking the iteration we have that the obtained set X satisfies $\text{ctw}(D - X) \leq c$. On the other hand, each of the removed $(c + 1)$ -cutwidth-minimal induced subdigraphs P had to

include at least one vertex from Y . Since each such P has at most $\mathcal{O}(c^2)$ vertices by Theorem 9, we conclude that $|X| \leq \mathcal{O}(c^2) \cdot |Y|$. \square

Next we show that Theorem 9 combined with the Sunflower Lemma approach to kernelization yields a polynomial kernel for c -CUTWIDTH VERTEX DELETION of size $k^{\mathcal{O}(c^2)}$. We will use the following variant of the Sunflower Lemma due to Fomin et al. [12], which is tailored to applications in kernelization. Here, a set Z is a *hitting set* of a set family \mathcal{F} if every set from \mathcal{F} has a nonempty intersection with Z . Also, Z is a *minimal hitting set* of \mathcal{F} if every its proper subset is not a hitting set of \mathcal{F} .

Lemma 8 (Lemma 3.2 of [12]). *Let d be a fixed integer. Let \mathcal{F} be a family of subsets of some universe U , each of cardinality at most d . Then, given an integer k , one can in time $\mathcal{O}(|\mathcal{F}| \cdot (k + |\mathcal{F}|))$ compute a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $|\mathcal{F}'| \leq d!(k + 1)^d$ such that the following holds: every subset $Z \subseteq U$ of size at most k is a minimal hitting set for \mathcal{F} if and only if it is a minimal hitting set for \mathcal{F}' .*

Theorem 12. *Let c be a fixed integer. There exists an algorithm that given an instance (D, k) of c -CUTWIDTH VERTEX DELETION where D is a semi-complete digraph on n vertices, runs in time $n^{\mathcal{O}(c^2)}$ and returns an equivalent instance (D', k) of c -CUTWIDTH VERTEX DELETION where D' is an induced subdigraph of D on at most $k^{\mathcal{O}(c^2)}$ vertices.*

Proof. Let $d = \mathcal{O}(c^2)$ be the upper bound on the number of vertices of any $(c + 1)$ -cutwidth-minimal semi-complete digraph, given by Theorem 9. Find all $(c + 1)$ -cutwidth-minimal semi-complete digraphs in D by iterating through all subsets A of vertices of size at most d , and verifying whether $D[A]$ is $(c + 1)$ -cutwidth-minimal. For a given set A , this can be done by applying the algorithm of Fomin and Pilipczuk [11, 21] $|X| + 1$ times: to check whether $\text{ctw}(D[A]) > c$ and $\text{ctw}(D[A \setminus \{a\}]) \leq c$ for each $a \in A$. Since we inspect only induced subdigraphs of size at most d , this takes time $2^{\mathcal{O}(\sqrt{c \log c})} \cdot n^d$ in total.

Let \mathcal{F} be the obtained family of vertex sets of $(c + 1)$ -cutwidth-minimal induced subdigraphs of D . It is clear that for any $X \subseteq V(G)$, we have that $\text{ctw}(D[X]) \leq c$ if and only if X does not fully contain any set A from \mathcal{F} . Conversely, a set Z is a deletion set to cutwidth at most c in D if and only if Z is a hitting set for \mathcal{F} .

Apply the algorithm of Lemma 8 to the family \mathcal{F} , yielding a subfamily \mathcal{F}' of size at most $d! \cdot k^{d+1}$ that has exactly the same minimal hitting sets of size at most k . Let $W \subseteq V(G)$ be the union of all the sets in \mathcal{F}' ; then $|W| \leq d \cdot |\mathcal{F}'| \leq d \cdot d! \cdot k^{d+1}$. Denote $D' = D[W]$. We claim that the instance (D, k) is a yes-instance of c -CUTWIDTH VERTEX DELETION if and only if (D', k) does; note that then D' can be output by the algorithm. The left-to-right implication is trivial: intersecting with W maps every solution to (D, k) to a non-larger solution to (D', k) .

For the converse implication, suppose $Z \subseteq W$ is such that $|Z| \leq k$ and $\text{ctw}(D' - Z) \leq c$. In particular, Z is a hitting set of \mathcal{F}' , so let $\hat{Z} \subseteq Z$ be any minimal hitting set of \mathcal{F}' contained in Z . Then $|\hat{Z}| \leq k$ and by Lemma 8 we infer that \hat{Z} is a minimal hitting set for \mathcal{F} as well. Since hitting sets for \mathcal{F} are exactly deletions sets to cutwidth at most c in D , we infer that $\text{ctw}(D - \hat{Z}) \leq c$ and (D, k) is a yes-instance. \square

We remark that for tournaments, we may plug in the bound of Theorem 8 instead of the

bound of Theorem 9. This yields a more explicit running time of $(2c + 2\lceil\sqrt{2c}\rceil + 1)^k \cdot n^{\mathcal{O}(1)}$ for the FPT algorithm, and reduces the approximation factor to $\mathcal{O}(c)$ and the kernel size to $k^{\mathcal{O}(c)}$.

Finally, we prove that the asymptotic running time of the algorithm of Theorem 10 is likely to be tight: under ETH, there is no algorithm with running time subexponential in k . This follows from an adaptation of the standard reduction from the VERTEX COVER problem parameterized by solution size to FEEDBACK VERTEX SET.

Theorem 13. *For each fixed nonnegative integer c , the c -CUTWIDTH VERTEX DELETION problem is NP-hard on tournaments. Moreover, unless ETH fails, it does not admit a $2^{o(n)}$ - or a $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ -time algorithm in this setting.*

Proof. Fix a nonnegative integer c . We show a reduction from VERTEX COVER parameterized by solution size to the c -CUTWIDTH VERTEX DELETION problem with a linear blow-up of the instance size, and we ensure that the instance produced by the reduction is a tournament. It is well-known (see e.g. [7]) that VERTEX COVER has no $2^{o(n+m)}$ algorithm under ETH, where n and m denote the number of vertices and edges of the input graph, respectively. Thus, the reduction will refute, under ETH, the existence of a $2^{o(n)}$ -time algorithm for c -CUTWIDTH VERTEX DELETION, so in particular a $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ -time parameterized algorithm will be refuted.

For the reduction, we will need a bounded-size tournament with cutwidth exactly c . To show that such a tournament exists for each $c \in \mathbb{N}$, we propose the family of *circular tournaments* (see Figure 5) defined as follows. Given two nonnegative integers t and x , the circular tournament $C_{t,x}$ has vertex set $V_t = \{v_1, v_2, \dots, v_{2t+1}\}$ and arc set $E_{t,x}$ such that for all $i, j \in [1, 2t+1]$ with $i < j$, we have:

- if $j \in [i+1, 2t]$, then $(v_i, v_j) \in E_{t,x}$ whenever $j - i \leq t$, otherwise $(v_j, v_i) \in E_{t,x}$;
- if $j = 2t+1$, then $(v_i, v_j) \in E_{t,x}$ whenever $i \in [1, x]$ or $j - i \leq t$, otherwise $(v_j, v_i) \in E_{t,x}$.

Let t be the smallest integer such that $\frac{t(t+1)}{2} \geq c$ and $x = \frac{t(t+1)}{2} - c$; then $x < t$ by the minimality of t . It can be checked that then $(v_1, v_2, \dots, v_{2t+1})$ is a sorted vertex ordering of $C_{t,x}$, hence it is minimum by Lemma 1. This implies that $\text{ctw}(C_{t,x})$ is equal to the width of this ordering, which is $\frac{t(t+1)}{2} - x = c$.

Let (G, k) be a parameterized instance of VERTEX COVER, we construct the parameterized instance $(T(G), k)$ of c -CUTWIDTH VERTEX DELETION as follows. Without loss of generality, we may assume that the vertices in V are ordered, that is $V = \{v_1, v_2, \dots, v_n\}$. Then $T(G)$ is constructed as follows:

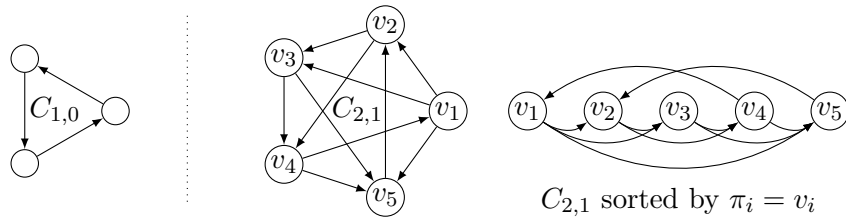


Fig. 5: The circular tournaments $C_{1,0}$ and $C_{2,1}$.

- For each $v_i \in V(G)$, add a copy $C_{t,x}^i$ of $C_{t,x}$ to $T(G)$ plus a fresh vertex v'_i ; denote $V^i = V(C_{t,x}^i)$. Moreover, for each vertex $u \in V^i$, add an arc (v'_i, u) to $T(G)$, so that v'_i has all the vertices of $C_{t,x}^i$ as outneighbors.
- For each $1 \leq i < j \leq n$, for every $u_i \in V^i$ and $u_j \in V^j$, add (v'_i, u_j) , (u_i, u_j) and (u_i, v'_j) to $T(G)$. Also, if $(v_i, v_j) \in E(G)$ then add (v'_j, v'_i) to $T(G)$, and otherwise add (v'_i, v'_j) .

Clearly, $T(G)$ is a tournament with a linear number of vertices, which can be computed in polynomial time. We prove that (G, k) is a yes-instance of VERTEX COVER if and only if $(T(G), k)$ is a yes-instance of c -CUTWIDTH VERTEX DELETION.

On one hand, let W' be a solution to the instance $(T(G), k)$ of c -CUTWIDTH VERTEX DELETION. We construct the solution W of (G, k) for VERTEX COVER as follows: if there exists $v \in (\{v'_i\} \cup V^i) \cap W'$ then put v_i into W . Clearly, $|W'| \leq |W| \leq k$. For the sake of a contradiction, suppose that W is not a vertex cover of G . Let $v_i v_j$ be an edge of G which is not covered by W . Then, the tournament $T(G) - W'$ still contains the induced subtournament $T_{i,j} = T[V^i \cup \{v'_i, v'_j\}]$. By construction, we have $d_{T_{i,j}}^-(v'_i) = 1$, $d_{T_{i,j}}^-(v'_j) = |V^i|$ and for every $u \in V^i$, we have $d_{T_{i,j}}^-(u) = 1 + d_{C_{t,x}^i}^-(u) \leq |V^i|$. Therefore, there exists a sorted ordering π of $T_{i,j}$ such that $\pi_1 = v'_i$ and $\pi_{|V(T_{i,j})|} = v'_j$. Also, since for every $u \in V^i$ we have $d_{T_{i,j}}^-(u) = 1 + d_{C_{t,x}^i}^-(u)$, the vertices of V^i are sorted in π as in a sorted ordering of $C_{t,x}^i$, so due to the edge (v'_j, v'_i) we have that $\text{ctw}(T_{i,j}, \pi) = 1 + \text{ctw}(C_{t,x}^i)$. By Lemma 1, we infer a contradiction:

$$\text{ctw}(T(G) - W') \geq \text{ctw}(T_{i,j}) = \text{ctw}(T_{i,j}, \pi) = 1 + \text{ctw}(C_{t,x}^i) = 1 + c.$$

On the other hand, let W be a solution of (G, k) for the VERTEX COVER problem. We naturally infer a solution W' to the instance $(T(G), k)$ of c -CUTWIDTH VERTEX DELETION as follows: whenever $v_i \in W$, we put v'_i into W' . By construction, for each $1 \leq i < j \leq n$, if there exists an arc from $u_j \in \{v'_j\} \cup V^j$ to $u_i \in \{v'_i\} \cup V^i$ in $T(G)$, then $u_j = v'_j$, $u_i = v'_i$ and $v_i v_j \in E(G)$. We deduce that either $v_i \in W$ and $v'_i \in W'$, or $v_j \in W$ and $v'_j \in W'$. In both cases, the arc (v'_j, v'_i) is removed when deleting W' from $T(G)$. Therefore, the remaining digraph $T(G) - W'$ contains no arc from $(\{v'_j\} \cup V^j) \setminus W'$ to $(\{v'_i\} \cup V^i) \setminus W'$, and it is easy to see that the digraph $T(G) - W'$ breaks into multiple strongly connected components, each either consisting of a single vertex v'_i , or being one of the gadgets $C_{t,x}^i$. Each of these strongly connected components has cutwidth at most c , so it follows that $\text{ctw}(T(G) - W') \leq c$. \square

6 Lower bounds

In this section, we prove almost tight lower bounds for the complexity of computing the cutwidth and the OLA-cost of a semi-complete digraph. Precisely, we prove the following result.

Theorem 14. *For semi-complete digraphs, both computing the cutwidth and computing the OLA-cost are NP-hard problems. Moreover, unless the Exponential Time Hypothesis fails:*

- *the cutwidth cannot be computed in time $2^{o(n)}$ nor in time $2^{o(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$; and*
- *the OLA-cost cannot be computed in time $2^{o(n)}$ nor in time $2^{o(k^{1/3})} \cdot n^{\mathcal{O}(1)}$.*

Here, n is the number of vertices of the input semi-complete digraph, and k is the target width/cost.

The proof of Theorem 14 spans the whole remainder of this section. We start our reduction from an instance of the NAE-3SAT problem, which was defined in Section 2 and for which a complexity lower bound under ETH is given by Corollary 1.

Let us introduce some notation. For a formula φ in CNF, the variable and clause sets of φ are denoted by $\text{vars}(\varphi)$ and $\text{cls}(\varphi)$, respectively. A variable assignment $\alpha: \text{vars}(\varphi) \rightarrow \{\perp, \top\}$ *NAE-satisfies* φ if every clause of φ has at least one, but not all literals satisfied. Formula φ is *NAE-satisfiable* if there is a variable assignment α that NAE-satisfies it; equivalently, both α and its negation $\neg\alpha$ satisfy φ . A digraph is called *basic* if it is simple and has no pair of symmetric arcs. For an integer $m > 0$, let λ_m be the following function $(14m + 1)$ -tuple:

$$\lambda_m(i) = \begin{cases} 2i, & \text{when } i \in [0, 5m] \\ 5m + i, & \text{when } i \in [5m + 1, 6m] \\ 11m, & \text{when } i \in [6m + 1, 7m] \\ 18m - i, & \text{when } i \in [7m + 1, 12m] \\ 42m - 3i, & \text{when } i \in [12m + 1, 14m] \end{cases}$$

The following lemma encapsulates the first, main step of our reduction.

Lemma 9. *There exists a polynomial-time algorithm that, given a 3CNF formula φ with m clauses, returns a basic digraph $D(\varphi)$ with $14m$ vertices and $24m$ edges such that:*

1. *for every vertex ordering π , we have $\text{cuts}(D(\varphi), \pi) \preceq \lambda_m$;*
2. *if φ is NAE-satisfiable, then there exists a vertex ordering π with $\text{cuts}(D(\varphi), \pi) = \lambda_m$;*
3. *if there is a vertex ordering π with $\max\{\text{cuts}(D(\varphi), \pi)\} \geq 11m$, then φ is NAE-satisfiable.*

Proof. Without loss of generality, we may assume that each clause of φ contains exactly 3 literals, by repeating some literal if necessary. Then, we may also assume that every variable of $\text{vars}(\varphi)$ appears at least twice, because a variable that appears only once can always be set in order that the clause in which it appears is NAE-satisfied, and thus such a variable and its associated clause may be safely removed. For every variable $x \in \text{vars}(\varphi)$, let p_x be the number of occurrences of x in the clauses of φ ; hence $3m = \sum_{x \in \text{vars}(\varphi)} p_x$ and $p_x \geq 2$ for each $x \in \text{vars}(\varphi)$.

We finally assume the clauses and literals are ordered, so we may say that a literal ℓ_x is the i_x th occurrence of variable x in the clauses of φ , with $i_x \in [1, p_x]$.

We now describe the construction of $D(\varphi)$; see Figure 6. For every variable $x \in \text{vars}(\varphi)$ construct a *variable gadget* G_x , which is a directed cycle of length $2p_x$ with vertices named as follows: $\perp_1^x \rightarrow \top_1^x \rightarrow \perp_2^x \rightarrow \top_2^x \rightarrow \dots \rightarrow \perp_{p_x}^x \rightarrow \top_{p_x}^x \rightarrow \perp_1^x$. Note that this cycle has no symmetric arcs since $p_x > 1$.

Then, for every clause $C \in \text{cls}(\varphi)$, where $C = \ell_x \vee \ell_y \vee \ell_z$ for literals of variables $x, y, z \in \text{vars}(\varphi)$, respectively, construct the following \top -clause gadget G_{\top}^C . Introduce a vertex \top^C and a set of vertices $V_{\top}^C = \{\top_{\ell_x}^C, \top_{\ell_y}^C, \top_{\ell_z}^C\}$ together with the following arcs:

- A directed 3-cycle $(\top_{\ell_x}^C, \top_{\ell_y}^C), (\top_{\ell_y}^C, \top_{\ell_z}^C), (\top_{\ell_z}^C, \top_{\ell_x}^C)$.
- A set $\{(\top^C, \top_{\ell_x}^C), (\top^C, \top_{\ell_y}^C), (\top^C, \top_{\ell_z}^C)\}$ of arcs from \top^C to the vertices of V_{\top}^C .

Similarly, construct the \perp -clause gadget G_{\perp}^C , which is isomorphic to G_{\top}^C , but with vertices named \perp . Gadgets G_{\top}^C and G_{\perp}^C will differ in how we connect them with the rest of the graph.

Intuitively, the variable assignment α , intended to NAE-satisfy φ , is encoded by choosing, in each variable gadget G_x , which vertices are placed in the first half of π , and which are placed

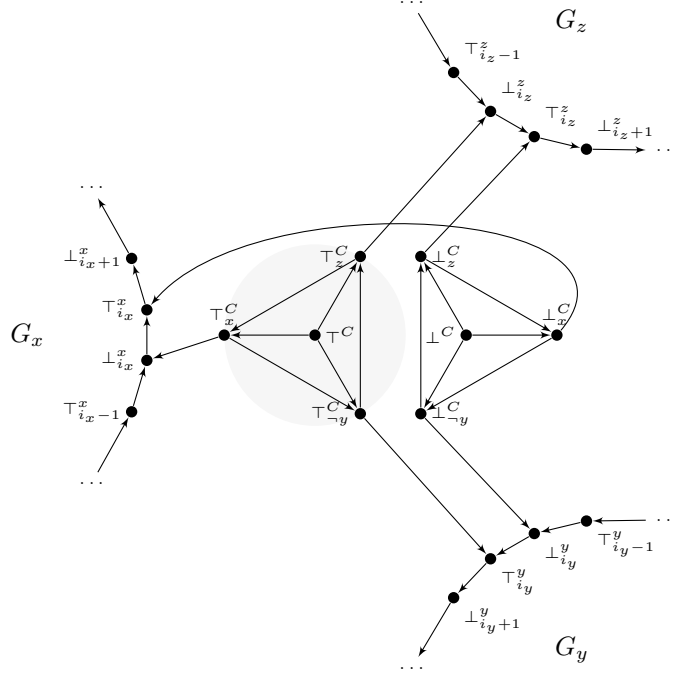


Fig. 6: Part of the construction of $D(\varphi)$. The \top -clause gadget G_{\top}^C (subgraph in the shaded circle) and the \perp -clause gadget G_{\perp}^C for $C = x \vee \neg y \vee z$, together with neighboring parts of variable gadgets G_x, G_y, G_z .

in the second. We use the gadget G_{\top}^C to verify that α satisfies C , whereas the gadget G_{\perp}^C verifies that $\neg\alpha$ also satisfies C . For this purpose, connect the clause gadgets to variable gadgets as follows. Suppose $\ell_x \in C$ is the i_x th occurrence of x . If $\ell_x = x$ then add two arcs $(\top_{\ell_x}^C, \perp_{i_x}^x)$ and $(\perp_{\ell_x}^C, \top_{i_x}^x)$, and if $\ell_x = \neg x$ then add two arcs $(\top_{\ell_x}^C, \top_{i_x}^x)$ and $(\perp_{\ell_x}^C, \perp_{i_x}^x)$.

This concludes the construction of $D(\varphi)$. Clearly $D(\varphi)$ is basic, and a straightforward verification using the equality $3m = \sum_{x \in \text{vars}(\varphi)} p_x$ shows that conditions $|V(D(\varphi))| = 14m$ and $|E(D(\varphi))| = 24m$ hold as well. We are left with verifying the three condition from the lemma statement. The following claim about the maximum size of a cut in $D(\varphi)$ will be useful.

Claim 3. *For any vertex subset $A \subseteq V$, we have that $|E(A, V \setminus A)| \leq 11m$.*

Proof. Denote $F = E(A, V \setminus A)$. First, consider any variable $x \in \text{vars}(\varphi)$. Since G_x is a directed cycle of length $2p_x$, it can easily be seen that $|F \cap E(G_x)| \leq p_x$ and the equality holds if and only if A contains every second vertex of the cycle G_x . Second, consider any clause $C = \ell_x \vee \ell_y \vee \ell_z \in \text{cls}(\varphi)$. Let R_{\top}^C be the set of three arcs connecting G_{\top}^C with the variable gadgets G_x, G_y , and G_z . Since \top^C has no incoming arcs, we can assume without loss of generality that $\top^C \in A$, as putting \top^C into A can only increase $|F| = |E(A, V \setminus A)|$. We now distinguish cases depending on the cardinality of $A \cap V_{\top}^C = A \cap \{\top_{\ell_x}^C, \top_{\ell_y}^C, \top_{\ell_z}^C\}$. The following implications follow from a straightforward analysis of the situation in G_{\top}^C and on incident arcs.

- If $|A \cap V_{\top}^C| = 0$ then $|F \cap R_{\top}^C| = 0$ and $|F \cap E(G_{\top}^C)| = 3$.
- If $|A \cap V_{\top}^C| = 1$ then $|F \cap R_{\top}^C| \leq 1$ and $|F \cap E(G_{\top}^C)| = 3$.

- If $|A \cap V_{\top}^C| = 2$ then $|F \cap R_{\top}^C| \leq 2$ and $|F \cap E(G_{\top}^C)| = 2$.
- If $|A \cap V_{\top}^C| = 3$ then $|F \cap R_{\top}^C| \leq 3$ and $|F \cap E(G_{\top}^C)| = 0$.

In all the cases, we conclude that $|F \cap (E(G_{\top}^C) \cup R_{\top}^C)| \leq 4$; note that the equality can hold only in the two middle ones. The same analysis applies to the \perp -clause gadgets, yielding $|F \cap (E(G_{\perp}^C) \cup R_{\perp}^C)| \leq 4$, where R_{\perp}^C is defined analogously. Since the set family

$$\{E(G_x) : x \in \text{vars}(\varphi)\} \cup \{(E(G_{\top}^C) \cup R_{\top}^C) \cup (E(G_{\perp}^C) \cup R_{\perp}^C) : C \in \text{cls}(\varphi)\}$$

forms a partition of the edge set of $D(\varphi)$, we get:

$$\begin{aligned} |F| &= \sum_{x \in \text{vars}(\varphi)} |F \cap E(G_x)| + \sum_{C \in \text{cls}(\varphi)} |F \cap (E(G_{\top}^C) \cup R_{\top}^C \cup E(G_{\perp}^C) \cup R_{\perp}^C)| \\ &\leq \sum_{x \in \text{vars}(\varphi)} p_x + \sum_{C \in \text{cls}(\varphi)} 8 = 11m, \end{aligned}$$

which finishes the proof of the claim. \lrcorner

We now verify the first condition from the lemma statement.

Claim 4. *For any vertex ordering π of $D(\varphi)$ and each position $i \in [0, 14m]$, we have*

$$\text{cuts}\langle D(\varphi), \pi \rangle(i) \leq \lambda_m(i). \quad (6)$$

Proof. Denote $V = V(D(\varphi))$ and $E = E(D(\varphi))$. Observe that the in-degree of each vertex is at most two by construction. Thus, for any vertex ordering π and each position i , we have

$$\text{cuts}\langle D(\varphi), \pi \rangle(i) \leq |E(V, \pi_{\leq i})| \leq 2i. \quad (7)$$

This verifies (6) for $i \in [0, 5m]$.

Observe that, every independent set in $D(\varphi)$ has at most $5m$ vertices: at most p_x vertices can be selected from each variable gadget G_x , for $x \in \text{vars}(\varphi)$, and at most 1 vertex from each clause gadget G_{\top}^C and G_{\perp}^C , for $C \in \text{cls}(\varphi)$. We deduce that every subset of i vertices in $D(\varphi)$ induces a subdigraph with at least $i - 5m$ arcs, and hence:

$$\text{cuts}\langle D(\varphi), \pi \rangle(i) = |E(V, \pi_{\leq i})| - |E(\pi_{\leq i}, \pi_{\leq i})| \leq 2i - (i - 5m) = 5m + i. \quad (8)$$

This verifies (6) for $i \in [5m + 1, 6m]$.

Reciprocally, the outdegree of each vertex of $D(\varphi)$ is at most three by construction, so

$$\text{cuts}\langle D(\varphi), \pi \rangle(i) \leq |E(\pi_{> i}, V)| \leq 3(|V| - i) = 42m - 3i. \quad (9)$$

This verifies (6) for $i \in [12m + 1, 14m]$.

To prove the next inequality, we make a finer analysis of vertices with outdegrees less than 3. For $1 \leq j \leq 3$ and a position i , let $V_i^j = \{u \in \pi_{> i} : d^+(u) = j\}$. Note that the set V_i^3 is a subset of the centers \top^C and \perp^C of the clause gadgets, thus $|V_i^3| \leq 2m$. We infer that

$$|E(\pi_{> i}, V)| = 3|V_i^3| + 2|V_i^2| + |V_i^1| = 2|\pi_{> i}| + |V_i^3| - |V_i^1| \leq 2(|V| - i) + 2m - |V_i^1|.$$

Also, any independent set in $D(\varphi)[V_i^2 \cup V_i^3]$ has size at most $2m$, since this digraph is an induced subdigraph of the union of the $2m$ clause gadgets. As before, we infer that this induced subdigraph has at least $|V_i^2| + |V_i^3| - 2m$ edges, hence

$$|E(\pi_{>i}, \pi_{>i})| \geq |V_i^2| + |V_i^3| - 2m = |V| - i - |V_i^1| - 2m.$$

Piecing this altogether, we deduce

$$\text{cuts}\langle D(\varphi), \pi \rangle(i) = |E(\pi_{>i}, V)| - |E(\pi_{>i}, \pi_{>i})| \leq |V| + 4m - i = 18m - i. \quad (10)$$

This verifies (6) for $i \in [7m + 1, 12m]$.

It remains to show the last inequality: for $i \in [6m + 1, 7m]$ it holds that

$$\text{cuts}\langle D(\varphi), \pi \rangle(i) \leq 11m. \quad (11)$$

This, however, follows immediately from applying Claim 3 to $A = \pi_{>i}$, so the proof of the claim is now complete. \lrcorner

It remains to prove the last two conditions of the lemma. We proceed with the third one.

Claim 5. *If there is a vertex ordering π with $\max\{\text{cuts}\langle D(\varphi), \pi \rangle\} \geq 11m$, then φ is NAE-satisfiable.*

Proof. Suppose there exist a vertex ordering π and position i such that $\text{cuts}\langle D(\varphi), \pi \rangle(i) \geq 11m$. By applying Claim 3 to $A = \pi_{>i}$ we infer that also $\text{cuts}\langle D(\varphi), \pi \rangle(i) \leq 11m$, so

$$|E_\pi^i| = |E(\pi_{>i}, \pi_{\leq i})| = \text{cuts}\langle D(\varphi), \pi \rangle(i) = 11m$$

and all inequalities used in the proof of Claim 3 for $A = \pi_{>i}$ are in fact equalities. In particular, from the examination of the proof it follows that:

- For every $x \in \text{vars}(\varphi)$, $\pi_{>i} \cap V(G_x)$ is either $\{\perp_x^i : i \in [1, p_x]\}$ or $\{\top_x^i : i \in [1, p_x]\}$.
- For every $C \in \text{cls}(\varphi)$, either $|\pi_{>i} \cap V_\top^C| = 1$ and $|E_\pi^i \cap R_\top^C| = 1$, or $|\pi_{>i} \cap V_\top^C| = 2$ and $|E_\pi^i \cap R_\top^C| = 2$. In any case, $|E_\pi^i \cap R_\top^C| \geq 1$, and similarly $|E_\pi^i \cap R_\perp^C| \geq 1$.

Let $\alpha : \text{vars}(\varphi) \rightarrow \{\perp, \top\}$ be an assignment defined as follows: for each $x \in \text{vars}(\varphi)$, set $\alpha(x) = \top$ if $\pi_{>i} \cap V(G_x) = \{\top_x^i : i \in [1, p_x]\}$, and set $\alpha(x) = \perp$ otherwise. We prove that α NAE-satisfies φ .

Take any $C \in \text{cls}(\varphi)$. Since $|E_\pi^i \cap R_\top^C| \geq 1$, there must exist an arc $(\top_{\ell_x}^C, \beta_i^x) \in R_\top^C$ such that $\top_{\ell_x}^C \in \pi_{>i}$ and $\beta_i^x \notin \pi_{>i}$. By the definition of α , the latter assertion is equivalent to $\alpha(x) = \neg\beta$. By the construction of $D(\varphi)$, the existence of an arc from $\top_{\ell_x}^C$ to β_i^x exactly means that ℓ_x is satisfied by setting $\alpha(x) = \neg\beta$. Since $|E_\pi^i \cap R_\perp^C| \geq 1$ as well, a symmetric reasoning shows that there also exists a literal of C which is not satisfied by α . Therefore, both α and $\neg\alpha$ satisfy φ , which means that α NAE-satisfies φ . \lrcorner

We are left with the second condition from the lemma statement.

Claim 6. *If φ is NAE-satisfiable, then there exists a vertex ordering π with $\text{cuts}\langle D(\varphi), \pi \rangle = \lambda_m$.*

Proof. Suppose that φ is NAE-satisfiable, and let $\alpha: \text{vars}(\varphi) \rightarrow \{\perp, \top\}$ be a NAE-satisfying assignment for φ . We construct a vertex ordering π such that $\text{cuts}(D(\varphi), \pi) = \lambda_m$ as follows. We list consecutive vertices in π from left to right, thinking of introducing them along the ordering. Whenever we do not specify the order of introducing some vertices, this order can be chosen arbitrarily.

- ($i \in [1, 3m]$) For each $x \in \text{vars}(\varphi)$, if $\alpha(x) = \top$, then introduce the vertices \perp_i^x of G_x , or introduce the vertices \top_i^x of G_x otherwise.
- ($i \in [3m + 1, 5m]$) For each clause $C \in \text{cls}(\varphi)$, introduce a vertex $\top_{\ell_x}^C \in V_{\top}^C$ and a vertex $\perp_{\ell_y}^C \in V_{\perp}^C$ such that α satisfies ℓ_y but does not satisfy ℓ_x . Since α NAE-satisfies φ , such vertices exist.
- ($i \in [5m + 1, 6m]$) Since we only introduced 2 vertices per clause so far, for each clause $C \in \text{cls}(\varphi)$ there remains an unused literal $\ell_z \in C$, i.e., a literal for which no corresponding vertex is already introduced. Introduce $\perp_{\ell_z}^C$ if α satisfies ℓ_z , or $\top_{\ell_z}^C$ otherwise.
- ($i \in [6m + 1, 7m]$) For each $C \in \text{cls}(\varphi)$, introduce $\top_{\ell_z}^C$ if α satisfies ℓ_z , or $\perp_{\ell_z}^C$ otherwise, where ℓ_z is as above.
- ($i \in [7m + 1, 9m]$) For each $C \in \text{cls}(\varphi)$, introduce $\perp_{\ell_x}^C$ and $\top_{\ell_y}^C$, where ℓ_x and ℓ_y are as three points above.
- ($i \in [9m + 1, 12m]$) For each $x \in \text{vars}(\varphi)$, if $\alpha(x) = \top$, introduce the vertices \top_i^x of G_x , or introduce the vertices \perp_i^x of G_x otherwise.
- ($i \in [12m + 1, 14m]$) For each $C \in \text{cls}(\varphi)$, introduce \top^C and \perp^C .

It is not hard to analyze the number of arcs that each introduced vertex π_i brings and removes, when moving from E_{π}^{i-1} to E_{π}^i . In fact, in the interval $[1, 5m]$ the cutsize is incremented by 2 with each new vertex, in $[5m + 1, 6m]$ it is incremented by 1, in $[6m + 1, 7m]$ it stays the same, in $[7m + 1, 12m]$ it decreases by 1, and in $[12m + 1, 14m]$ it decreases by 3. This shows that the cut vector of π is exactly equal to λ_m , as claimed. \lrcorner

Claims 4, 5, and 6 together finish the proof of Lemma 9. \square

Note that Lemma 9 expresses a reduction from NAE-3SAT to a maximization problem: NAE-satisfiability of φ is equivalent to $D(\varphi)$ admitting a vertex ordering of width *at least* $11m$. By complementing the resulting digraph, we turn this maximization into a minimization problem. Precisely, given a simple digraph $D = (V, E)$, define its *complement* as $\bar{D} = (V, \bar{E})$, where $\bar{E} = V^2 \setminus (E \cup \{(u, u) : u \in V\})$. That is, we take the complete digraph without self-loops on the vertex set V , and we remove all the arcs that are present in D . Note that the complement of a basic digraph is semi-complete.

Now, let $\bar{\lambda}_m$ be the tuple such that for all $i \in [0, 14m]$, we have $\lambda_m(i) + \bar{\lambda}_m(i) = i(14m - i)$. It is not hard to check that $\max\{\bar{\lambda}_m\} = \bar{\lambda}_m(7m) = 49m^2 - 11m$. A simple verification of how the conditions of Lemma 9 are transformed under complementation yields the following.

Lemma 10. *The complement of $D(\varphi)$ is a semi-complete digraph $\bar{D}(\varphi)$ satisfying:*

1. for every vertex ordering π , we have $\bar{\lambda}_m \preceq \text{cuts}(\bar{D}(\varphi), \pi)$;
2. if φ is NAE-satisfiable, then there exists a vertex ordering π with $\text{cuts}(\bar{D}(\varphi), \pi) = \bar{\lambda}_m$;
3. if $\bar{D}(\varphi)$ admits a vertex ordering π of width at most $49m^2 - 11m$, then φ is NAE-satisfiable.

Proof. Observe that for any digraph D on n vertices, any its vertex ordering π , and any $i \in [0, n]$, we have $\text{cuts}(D, \pi)(i) + \text{cuts}(\bar{D}, \pi)(i) = i(n - i)$, and then

$$\text{cuts}(D(\varphi), \pi)(i) + \text{cuts}(\bar{D}(\varphi), \pi)(i) = \lambda_m(i) + \bar{\lambda}_m(i). \quad (12)$$

Equality (12) implies that $\bar{\lambda}_m(i) \leq \text{cuts}(\bar{D}(\varphi), \pi)(i)$ is equivalent to $\text{cuts}(D(\varphi), \pi)(i) \leq \lambda_m(i)$. By the first claim of Lemma 9, we deduce that $\bar{\lambda}_m \preceq \text{cuts}(\bar{D}(\varphi), \pi)$ for every vertex ordering π . Similarly, if φ is NAE-satisfiable, then by the second claim of Lemma 9 we have an ordering π such that $\text{cuts}(D(\varphi), \pi) = \lambda_m$, and then $\text{cuts}(\bar{D}(\varphi), \pi) = \bar{\lambda}_m$ due to (12). Finally, if we have a vertex ordering π of width at most $49m^2 - 11m$, that is, with $\max\{\text{cuts}(\bar{D}(\varphi), \pi)\} = 49m^2 - 11m$, then in particular $\text{cuts}(\bar{D}(\varphi), \pi)(7m) \leq 49m^2 - 11m$, which is equivalent to $\text{cuts}(D(\varphi), \pi)(7m) \geq 11m$, due to (12). By the third claim of Lemma 9, this implies that φ is NAE-satisfiable. \square

Thus, Lemma 10 shows that NAE-satisfiability of φ is equivalent to $\bar{D}(\varphi)$ having cutwidth at most $49m^2 - 11m$. However, the fact that NAE-satisfiability of φ implies that $\bar{D}(\varphi)$ admits a vertex ordering with a very concrete cut vector $\bar{\lambda}_m$, which is the best possible in the sense of the first claim of Lemma 10, also enables us to derive a lower bound for OLA. All these observations, together with the linear bound on the number of vertices of $\bar{D}(\varphi)$, make the proof of Theorem 14 essentially complete. We now give a formal verification that Theorem 14 follows from Lemmas 9 and 10.

Proof of Theorem 14. Suppose we are given an instance φ of the NAE-3SAT problem. Compute the semi-complete digraph $\bar{D}(\varphi)$; this takes polynomial time, and $\bar{D}(\varphi)$ has $14m$ vertices. By Lemmas 9 and 10, we have that φ is NAE-satisfiable if and only if the cutwidth of $\bar{D}(\varphi)$ is at most $49m^2 - 11m$. Indeed, if φ is NAE-satisfiable, then $\bar{D}(\varphi)$ has an ordering with cut vector $\bar{\lambda}_m$, whose maximum is $49m^2 - 11m$, and the converse implication is exactly the third claim of Lemma 10. Therefore, if we could verify whether a given semi-complete digraph D on n vertices has cutwidth at most k in time $2^{o(n)}$ or $2^{o(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$, then by applying this algorithm to $\bar{D}(\varphi)$ we would resolve whether φ is NAE-satisfiable in time $2^{o(m)} \cdot (n + m)^{\mathcal{O}(1)}$, contradicting ETH by Corollary 1.

For OLA, we adapt the argument slightly. Let $k = \sum\{\bar{\lambda}_m\}$; note that $k \in \Theta(m^3)$. Again, by Lemmas 9 and 10 we infer that φ is NAE-satisfiable if and only if the OLA-cost of $\bar{D}(\varphi)$ is at most k . Indeed, if φ is NAE-satisfiable, then $\bar{D}(\varphi)$ has an ordering with cut vector $\bar{\lambda}_m$, whose sum is k . On the other hand, by the first claim of Lemma 10 we have that every vertex ordering π of $\bar{D}(\varphi)$ satisfies $\bar{\lambda}_m \preceq \text{cuts}(\bar{D}(\varphi), \pi)$, so the only possibility of obtaining an ordering of cost at most k is to have an ordering with cut vector equal to $\bar{\lambda}_m$. This cut vector would in particular have maximum equal to $49m^2 - 11m$, which by the third claim of Lemma 10 implies that φ is NAE-satisfiable. As before, this means that using an algorithm for computing OLA-cost with running time $2^{o(n)}$ or $2^{o(\sqrt[3]{k})} \cdot n^{\mathcal{O}(1)}$ we would be able to resolve NAE-satisfiability of φ in time $2^{o(m)} \cdot (n + m)^{\mathcal{O}(1)}$, contradicting ETH by Corollary 1. \square

7 Parameterization by the number of pure vertices

Consider the problems of computing the cutwidth and the OLA-cost of a semi-complete digraph. On one hand, the reduction of Lemma 9 constructs a basic digraph whose complement has a pair of symmetric arcs between almost every pair of vertices. On the other hand, on tournaments the problems are polynomial-time solvable. These observations suggest taking a closer look at the parameterization by the number of vertices incident to symmetric arcs, as this parameter is zero in the tournament case and high in instances constructed in our hardness reduction. We indeed show that this parameterization leads to an FPT problem, even in a larger generality. Call a vertex u of a simple digraph D *pure* if for any other vertex v , exactly one of the arcs (u, v) or (v, u) is present in D . In this section we prove the following algorithmic result.

Theorem 15. *There is an algorithm that, given a simple digraph D on n vertices, computes the cutwidth and the OLA-cost of D in time $2^k \cdot n^{\mathcal{O}(1)}$, where k is the number of non-pure vertices in D . The algorithm can also report orderings certifying the output values.*

The proof of Theorem 15 is based on the fine understanding of minimum orderings that we developed in Section 3. Recall that in Lemma 1 we have shown that in the case when all vertices are pure—that is, when the digraph is a tournament—minimum orderings are exactly the sorted ones. We now extend this observation to the case when not all vertices are pure, by showing that at least the pure ones may be sorted greedily according to their indegrees.

Lemma 11. *Let P be the set of pure vertices of a simple digraph D , and let π^P be a sorted ordering of P , that is, for each $u, v \in P$, if $d_D^-(u) < d_D^-(v)$ then $\pi^P(u) < \pi^P(v)$. There exists a ctw -optimal ordering π of D and an OLA-optimal ordering π' of D such that the restrictions of π and π' to P are equal to π^P .*

Proof. We only give the proof for cutwidth as the proof for OLA follows directly by changing each occurrence of ctw by OLA. Let σ be an arbitrary ctw -optimal vertex ordering of D . Suppose that the restriction σ^P of σ to P differs from π^P . We show how to transform σ into an ctw -optimal ordering π of D such that the restriction of π to P is π^P .

Let j be the smallest index such that $\pi_j^P \neq \sigma_j^P$. Let $u = \pi_j^P$ and $v = \sigma_j^P$. By the minimality of j , the prefixes of π^P and σ^P up to position $j - 1$ match, so v must be later than u in π^P , but v is before u in σ^P . Since σ^P is equal to σ restricted to P , we have that v is before u in σ . On the other hand, since π^P is sorted with respect to indegrees in D , from $\pi^P(u) < \pi^P(v)$ we infer the inequality

$$d^-(u) \leq d^-(v). \quad (13)$$

We define a vertex ordering σ^* of D as a copy of σ except that we exchange the position of vertices u and v . Note that σ^* and π match on a prefix of length i , one longer than on which σ and π matched. We claim that $\text{ctw}(D) = \text{ctw}(D, \sigma) = \text{ctw}(D, \sigma^*)$, that is, σ^* is a ctw -optimal ordering of D . Note that if this was the case, then we could apply the same reasoning to σ^* and further on, eventually obtaining an ctw -optimal ordering of D whose restriction to P is exactly equal to π^P . This would conclude the proof.

It remains to prove the claim. Recall that $\sigma(v) < \sigma(u)$. By definition of a cut vector, for all $i \notin [\sigma(v), \sigma(u)]$, we have $\text{cuts}(D, \sigma)(i) = \text{cuts}(D, \sigma^*)(i)$. Let us consider the cuts at positions

$i \in [\sigma(v), \sigma(u)]$. We have:

$$\begin{aligned} \text{cuts}\langle D, \sigma \rangle(i) &= |E(\sigma_{>i}, \sigma_{\leq i})| \\ &= |E(V, \sigma_{\leq i})| - |E(\sigma_{\leq i}, \sigma_{\leq i})| \\ &= |E(V, \sigma_{\leq i} \setminus \{v\})| + d^-(v) - |E(\sigma_{\leq i}, \sigma_{\leq i})| \end{aligned} \quad (14)$$

The same reasoning for σ^* yields:

$$\text{cuts}\langle D, \sigma^* \rangle(i) = |E(V, \sigma_{\leq i}^* \setminus \{u\})| + d^-(u) - |E(\sigma_{\leq i}^*, \sigma_{\leq i}^*)| \quad (15)$$

Also, for any pure vertex w belonging to any vertex set W , we have:

$$\begin{aligned} |E(W, W)| &= |E(W \setminus \{w\}, W \setminus \{w\})| + d_{D[W]}^+(w) + d_{D[W]}^-(w) - |E(w, w)| \\ &= |E(W \setminus \{w\}, W \setminus \{w\})| + |W| - 1. \end{aligned}$$

Since u and v are pure vertices and $\sigma_{\leq i} \setminus \{v\} = \sigma_{\leq i}^* \setminus \{u\}$, it follows that:

$$\begin{aligned} |E(\sigma_{\leq i}, \sigma_{\leq i})| &= |E(\sigma_{\leq i} \setminus \{v\}, \sigma_{\leq i} \setminus \{v\})| + |\sigma_{\leq i} \setminus \{v\}| - 1 \\ &= |E(\sigma_{\leq i}^* \setminus \{u\}, \sigma_{\leq i}^* \setminus \{u\})| + |\sigma_{\leq i}^* \setminus \{u\}| - 1 = |E(\sigma_{\leq i}^*, \sigma_{\leq i}^*)| \end{aligned}$$

By combining this with (14) and (15), we conclude that

$$\text{cuts}\langle D, \sigma \rangle(i) - \text{cuts}\langle D, \sigma^* \rangle(i) = d^-(v) - d^-(u),$$

which is a positive value by (13).

We conclude that $\text{cuts}\langle D, \sigma^* \rangle(i) \leq \text{cuts}\langle D, \sigma \rangle(i)$ for each $i \in [0, |V(D)|]$. Thus, we deduce that $\text{ctw}(D) \leq \text{ctw}(D, \sigma^*) \leq \text{ctw}(D, \sigma) = \text{ctw}(D)$, so σ^* is indeed a ctw -optimal vertex ordering of D . This concludes the proof. \square

We can now proceed to the proof of Theorem 15.

Proof of Theorem 15. We focus on computing the cutwidth of D , at the end we will briefly argue how the algorithm can be adjusted for the OLA-cost. Let P and Q be the sets of pure and non-pure vertices in D , respectively. Compute any ordering ν^P of P that sorts the vertices of P according to non-decreasing indegrees in D . By Lemma 11, we know that there exists a ctw -optimal vertex ordering π of D such that the restriction of π to P is equal to ν^P . We give a dynamic programming algorithm that attempts at reconstructing π based on ν^P .

The set space of the dynamic programming consists of pairs (X, i) , where X is a subset of Q and i is an integer with $0 \leq i \leq |P|$. Thus, we have at most $2^k \cdot (n+1)$ states. For a state (X, i) , let $S(X, i) = X \cup \nu^P[i]$ be the associated candidate for a prefix of π . We define the following value function for the states:

$$\phi(X, i) = \min_{\sigma: \text{ordering of } S(X, i)} \left(\max_{j=1, 2, \dots, i} \{|E(V(D) \setminus \sigma[j], \sigma[j])|\} \right).$$

In other words, assuming that $S(X, i)$ is a prefix of the constructed ordering π , $\phi(X, i)$ tells us how small maximum cutsize we can obtain among cuts within this prefix. It is straightforward

to verify that function $\phi(X, i)$ satisfies the following recurrence, which corresponds to choosing whether the last vertex of the ordering of $S(X, i)$ belongs to X or to $\nu^P[i]$.

$$\begin{aligned}\phi(\emptyset, 0) &= 0 \\ \phi(X, i) &= \max(|E(V(D) \setminus S(X, i), S(X, i))|, \\ &\quad \min(\{\phi(X, i-1)\} \cup \{\phi(X \setminus \{x\}, i) : x \in X\})).\end{aligned}$$

Here, we use the convention that $\phi(X, -1) = +\infty$ for all $X \subseteq Q$. Thus, we can compute the values of function $\phi(\cdot, \cdot)$ in the dynamic programming manner, by iterating on sets X of increasing size and increasing i . The computation of each value takes polynomial time and there are $2^k \cdot n$ values to compute, hence the running time follows. The optimum value of cutwidth can be found as the value $\phi(Q, |P|)$, and an ordering certifying this value can be recovered in polynomial time using the standard method of back-links. To see that the algorithm is correct, observe that since π restricted to P is equal to ν^P , it follows that π gives rise to a computation path of the dynamic programming above that results in finding the optimum value of the cutwidth.

To adjust the algorithm to computing the OLA-cost of D , consider the adjusted value function for the states:

$$\psi(X, i) = \min_{\sigma: \text{ordering of } S(X, i)} \left(\sum_{j=1}^i |E(V(D) \setminus \sigma[j], \sigma[j])| \right).$$

Then, $\psi(\cdot, \cdot)$ satisfies the following recurrence:

$$\begin{aligned}\psi(\emptyset, 0) &= 0 \\ \psi(X, i) &= |E(V(D) \setminus S(X, i), S(X, i))| + \min(\{\psi(X, i-1)\} \cup \{\psi(X \setminus \{x\}, i) : x \in X\}).\end{aligned}$$

Thus, the values of $\psi(\cdot, \cdot)$ can be computed in the same manner within the same time complexity. Again, the OLA-cost of D is equal to $\psi(Q, |P|)$ and the ordering certifying this value can be recovered in polynomial time. The argument for the correctness is the same. \square

8 Conclusions

In this work we have charted the computational complexity of cutwidth and OLA on semi-complete digraphs by proving almost tight algorithmic lower bounds under ETH and showing that cutwidth admits a quadratic Turing kernel, even though a classic polynomial kernel cannot be expected unless $\text{NP} \subseteq \text{coNP}/\text{poly}$. A particular question that we leave open is whether the size of our Turing kernel for cutwidth could be improved to linear. This might be suggested by the linear bound on the size of a c -cutwidth minimal tournament (Theorem 8).

Another interesting direction is to investigate further the complexity of graph modification problems related to cutwidth: apply at most k modifications to the given semi-complete digraph in order to obtain a digraph of cutwidth at most c . Some immediate corollaries for vertex deletions are discussed in Section 5, but it is also interesting to look at the arc reversal variant, where the allowed modification is reversing an arc. For $c = 0$, this problem is equivalent to the FEEDBACK ARC SET problem, which has been studied intensively in tournaments and

semi-complete digraphs [1, 4, 10, 11]. Further results on the vertex deletion and arc reversal problems related to cutwidth in semi-complete digraphs will be the topic of a future paper, currently under preparation.

References

- [1] N. Alon, D. Lokshtanov, and S. Saurabh. Fast FAST. In *ICALP 2009*, volume 5555 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2009.
- [2] J. Bang-Jensen and G. Gutin. *Digraphs — theory, algorithms and applications*. Springer, 2002.
- [3] D. Binkele-Raible, H. Fernau, F. V. Fomin, D. Lokshtanov, S. Saurabh, and Y. Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. *ACM Transactions on Algorithms*, 8(4):38, 2012.
- [4] I. Bliznets, M. Cygan, P. Komosa, L. Mach, and M. Pilipczuk. Lower bounds for the parameterized complexity of Minimum Fill-in and other completion problems. In *SODA 2016*, pages 1132–1151. SIAM, 2016.
- [5] M. Chudnovsky, A. Fradkin, and P. D. Seymour. Tournament immersion and cutwidth. *Journal of Combinatorial Theory, Series B*, 102(1):93 – 101, 2012.
- [6] M. Chudnovsky and P. Seymour. A well-quasi-order for tournaments. *Journal of Combinatorial Theory, Series B*, 101(1):47 – 53, 2011.
- [7] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [8] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- [9] A. Drucker. New limits to classical and quantum instance compression. *SIAM Journal of Computing*, 44(5):1443–1479, 2015.
- [10] U. Feige. Faster FAST (Feedback Arc Set in Tournaments). *CoRR*, abs/0911.5094, 2009.
- [11] F. V. Fomin and M. Pilipczuk. Subexponential parameterized algorithm for computing the cutwidth of a semi-complete digraph. In *ESA 2013*, volume 8125 of *Lecture Notes in Computer Science*, pages 505–516. Springer, 2013.
- [12] F. V. Fomin, S. Saurabh, and Y. Villanger. A polynomial kernel for Proper Interval Vertex Deletion. *SIAM Journal on Discrete Mathematics*, 27(4):1964–1976, 2013.
- [13] A. Fradkin. *Forbidden structures and algorithms in graphs and digraphs*. PhD thesis, Princeton University, 2011.
- [14] A. Fradkin and P. D. Seymour. Edge-disjoint paths in digraphs with bounded independence number. *Journal of Combinatorial Theory, Series B*, 110:19–46, 2015.

- [15] V. Garnero and M. Weller. Parameterized certificate dispersal and its variants. *Theoretical Computer Science*, 622:66–78, 2016.
- [16] A. C. Giannopoulou, M. Pilipczuk, J. Raymond, D. M. Thilikos, and M. Wrochna. Cutwidth: Obstructions and algorithmic aspects. In *IPEC 2016*, volume 63 of *LIPICs*, pages 15:1–15:13. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2016.
- [17] D. Hermelin, S. Kratsch, K. Sołtys, M. Wahlström, and X. Wu. A completeness theory for polynomial (Turing) kernelization. *Algorithmica*, 71(3):702–730, 2015.
- [18] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
- [19] B. M. P. Jansen. Turing kernelization for finding long paths and cycles in restricted graph classes. *Journal of Computer and System Sciences*, 85:18–37, 2017.
- [20] M. Pilipczuk. Computing cutwidth and pathwidth of semi-complete digraphs via degree orderings. In *STACS 2013*, volume 20 of *LIPICs*, pages 197–208. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 2013.
- [21] M. Pilipczuk. *Tournaments and optimality: new results in parameterized complexity*. PhD thesis, University of Bergen, Norway, 2013.
- [22] A. Schäfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.
- [23] T. J. Shaefer. The complexity of satisfiability problems. In *STOC 1978*, pages 216–226. ACM, 1978.
- [24] R. Thomas. A Menger-like property of tree-width: the finite case. *Journal of Combinatorial Theory Series B*, 48(1):67–76, 1990.
- [25] S. Thomassé, N. Trotignon, and K. Vušković. A polynomial Turing-kernel for Weighted Independent Set in bull-free graphs. *Algorithmica*, 77(3):619–641, 2017.