Exploring the Complexity of Layout Parameters in Tournaments and Semi-Complete Digraphs[*][†]

Florian Barbero[1], Christophe Paul[2], and Michał Pilipczuk[3]

1   **LIRMM, Université de Montpellier, Montpellier, France**
     `florian.barbero@lirmm.fr`
2   **LIRMM, CNRS, Université de Montpellier, Montpellier, France**
     `christophe.paul@lirmm.fr`
3   **University of Warsaw, Warsaw, Poland**
     `michal.pilipczuk@mimuw.edu.pl`

──── **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 NP ⊆ coNP/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.

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 a rich and interesting subclass of directed graphs; we refer to the book of Bang-Jensen and Gutin [1] 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 $\mathsf{ctw}(D)$, 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 [3, 4, 10], who identified this layout parameter as the right 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 [3, 16]. 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 [4].

The algorithmic properties of cutwidth were preliminarily investigated by Chudnovsky, Fradkin, and Seymour [3, 10, 9]. In Fradkin's PhD thesis [9], 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 $2^{\mathcal{O}(\sqrt{k \log k})} \cdot n^2$ [8, 17], 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 $2^{\mathcal{O}(k^{1/3}\sqrt{\log k})} \cdot n^2$ [8, 17], where k is the target cost. It was left open whether the running times of these parameterized algorithms are optimal [17]. In fact, even settling the NP-hardness of computing cutwidth and OLA in semi-complete digraphs was open [9, 17].

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). Second, we describe the kernelization complexity of the two parameters in semi-complete digraphs. In particular, we show, somewhat surprisingly, that the problem of computing the cutwidth admits a quadratic Turing kernel, while the existence of a classic polynomial kernel would imply that $\text{NP} \subseteq \text{coNP}/\text{poly}$. The proofs of these main results yield complementary algorithmic and structural results that we discuss later.

Our algorithmic lower bounds are encapsulated in the following theorem.

▶ **Theorem 1.** *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 vertex count of the input semi-complete digraph, and k is the target width/cost.

Thus, Theorem 1 shows that the known parameterized algorithms of Fomin and Pilip-czuk [8] are optimal under ETH, up to $\sqrt{\log k}$ factor in the exponent. Note that both cutwidth and OLA 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.

Next, we turn our attention to kernelization. Recall that 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 [6, 5]. Consider the parameterized problems of deciding whether a given semi-complete digraph has cutwidth, respectively OLA-cost, bounded by a given integer c, which is considered to be the parameter. As shown by the next two theorems, the kernelization complexity of these two problems is quite different.

▶ **Theorem 2.** *There exists a polynomial-time algorithm that given an arbitrary digraph D and an integer c, either correctly concludes that $\mathrm{OLA}(D) > c$, or finds a digraph D' on at most $2c$ vertices such that $\mathrm{OLA}(D') = \mathrm{OLA}(D)$.*

▶ **Theorem 3.** *Unless $\mathsf{NP} \subseteq \mathsf{coNP}/poly$, there exists no polynomial-size kernelization algorithm for the problem of computing the cutwidth of a semi-complete digraph.*

The proofs of these two theorems 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 Theorem 2. On the other side, we can observe that the cutwidth of a 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 [7, 5].

However, an alternative notion of kernelization, called *Turing kernelization*, has been also studied intensively in the literature; cf. the discussion in [5]. 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. Somewhat surprisingly, we prove that the problem of computing the cutwidth of a semi-complete digraph admits a quadratic Turing kernel, which is encapsulated in the following theorem.

▶ **Theorem 4.** *There exists a polynomial-time algorithm that given a semi-complete digraph D and integer c, either correctly concludes that $\mathsf{ctw}(D) > c$ or outputs a list of at most n induced subdigraphs $D_1, \ldots, D_\ell$ of D, each with at most $\mathcal{O}(c^2)$ vertices, such that $\mathsf{ctw}(D) \le c$ if and only if $\mathsf{ctw}(D_i) \le c$ for each $i \in \{1, 2, \ldots, \ell\}$.*

Theorem 4 gives a so-called AND-*Turing kernel*, meaning that the algorithm 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 [2, 15, 11, 18, 21]. 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 [2, 11, 18], or adaptative kernels that fully exploit the oracle model [15, 21]. As separating classic and Turing kernelization is arguably one of the most important complexity-theoretical open problems within parameterized complexity [6, 13, 5], we find this new example intriguing.

As a byproduct of our approach to proving Theorem 4, we obtain also polynomial upper bounds on the sizes of minimal obstructions to having small cutwidth. More precisely, 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.

▶ **Theorem 5.** *For every positive integer c, every c-cutwidth-minimal semi-complete digraph has at most $\mathcal{O}(c^2)$ vertices.*

▶ **Theorem 6.** *For every positive integer c, every c-cutwidth-minimal tournament has at most $2c + 2\lceil\sqrt{2c}\rceil + 1$ vertices.*

The bound of Theorem 6 is almost tight, as there exist c-cutwidth-minimal tournaments with $2c+1$ vertices. Theorems 5 and 6 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.

Approach.   The starting point of our study is the approach used in the earlier works by Fradkin [9] and by the third author [17, 16], 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 [9], this statement may be made precise for tournaments: any indegree ordering has optimum cutwidth.

We present a somewhat finer study of this argument using the notion of a *minimum ordering*. Namely, a vertex ordering π of a digraph D is *minimum* if for any other vertex ordering π' of D and any $i \in \{1, 2, \ldots, n-1\}$, the cutsize in π between the prefix of length i and the complementary suffix is smaller or equal than the cutsize defined in the same manner in π'. The sorting argument of Fradkin [9] in fact yields the following: a vertex ordering of a tournament is minimum if and only if it is sorted according to indegrees. In particular, every tournament admits a minimum ordering, computable in polynomial time. Since every minimum ordering optimizes both the cutwidth and the OLA-cost, we obtain the following.

▶ **Theorem 7.** *The cutwidth and* OLA *of a tournament can be computed in polynomial time.*

Unfortunately, general semi-complete digraphs may not admit minimum orderings. However, a semi-complete digraph can be relaxed to a *fractional tournament* with a loss of factor 2 on the cutwidth and the OLA-cost. The ordering argument for tournaments may be applied to fractional tournaments as well, and thus we obtain a polynomial-time 2-approximation.

▶ **Theorem 8.** *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.*

While Theorems 7 and 8 for cutwidth were already proved by Fradkin [9], the applicability of the approach to OLA is a new contribution of this work. We choose to include the proofs of Theorems 7 and 8 in this work for two reasons. First, the fine understanding of minimum orderings is a basic tool needed in the proofs of our main results. Second, the abovementioned results of Fradkin [9] 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.

For the proof of Theorem 1, we construct a reduction from NAE-3SAT, a variant of 3SAT. In case the input formula is satisfiable, the output semi-complete digraph admits a minimum vertex ordering with a precisely specified vector of cutsizes. On the other hand, admitting any ordering of width bounded by the maximum of these cutsizes implies satisfiability of the input formula. Thus, the same reduction may serve to certify the hardness of both computing the cutwidth and computing the OLA-cost of a semi-complete digraph.

For the proof of Theorem 4, we use the notion of a *lean ordering*; see e.g. [12, 20]. 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 [4].

Lean orderings are used in the proof of Theorem 4 as follows. We first compute a 2-approximate ordering using Theorem 8, and then we exhaustively improve it until it becomes lean. 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.

Theorem 5 follows easily from Theorem 4; basically, the algorithm applied to a c-cutwidth minimal semi-complete digraph cannot output only smaller digraphs. For Theorem 6 we use our finer understanding of minimum orderings in tournaments. We remark that from the well-quasi order result of Chudnovsky and Seymour [4], it follows that 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.

Organization. In Section 2 we introduce notation, recall basic definitions, and prove Theorems 7 and 8. In Sections 3 and 4 we prove Theorems 4 and 1, respectively. The proofs of statements marked with ♠ will appear in the full version of the paper.

2 Preliminaries and basic results

Notation. We use standard graph notation for digraphs. 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 $\pi \colon V(D) \to [1, n]$, 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 $\pi_{\leq i} = \{\pi_j \colon j \in [1, i]\}$; 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 and concatenation of ordering(s) 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 $\mathsf{cuts}\langle D, \pi \rangle = (|E_\pi^0|, |E_\pi^1|, \ldots, |E_\pi^n|)$ is called the *cut vector* of π, and we denote $\mathsf{cuts}\langle D, \pi \rangle(i) = |E_\pi^i|$. 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 $\mathsf{cuts}\langle D, \pi \rangle \preceq \mathsf{cuts}\langle D, \pi' \rangle$. Note that a minimum vertex ordering may not exist.

The *width* of a vertex ordering π of a digraph D, denoted $\mathsf{ctw}(D, \pi)$, is equal to $\max\{\mathsf{cuts}\langle D, \pi \rangle\}$, where max on a tuple yields the largest coordinate. The *cutwidth* of D, denoted $\mathsf{ctw}(D)$, is the minimum width among vertex orderings of D. Similarly, the *cost* of π, denoted $\mathrm{OLA}(D, \pi)$, is equal to $\sum\{\mathsf{cuts}\langle D, \pi \rangle\}$, where $\sum$ on a tuple yields the sum of coordinates. This is equivalent to summing $j - i$ 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.

Exponential-Time Hypothesis. The *Exponential Time Hypothesis* (ETH) of Impagliazzo et al. [14] states that for some constant $c > 0$, there is no algorithm for 3SAT that would run in time $2^{cn} \cdot (n + m)^{\mathcal{O}(1)}$, where n and m are the numbers of variables and clauses of the input formula, respectively. Using the *Sparsification Lemma* [14] one can show that under ETH, there is a constant $c > 0$ such that 3SAT cannot be solved in time $2^{cm} \cdot (n + m)^{\mathcal{O}(1)}$. 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 [19] gave a linear reduction from 3SAT to NAE-3SAT, which immediately yields:

▶ **Corollary 9.** *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.*

Theorems 7 and 8. We now proceed to proving Theorems 7 and 8; recall that for cutwidth, these results have been already established by Fradkin [9]. However, we use this opportunity to present the reasoning in a more insightful manner and more general context, which also yields a better combinatorial understanding that will be helpful later. The core idea is to work in a more general setting of linear relaxations of tournaments, as defined next.

A *fractional tournament* is a pair $T = (V, \omega)$, where V is a finite vertex set and $\omega \colon V^2 \to \mathbb{R}_{\geq 0}$ 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 encapsulates the essence of our approach.

▶ **Lemma 10 (♠).** *A vertex ordering of a fractional tournament is minimum iff it is sorted.*

The proof of Theorem 7, even in the more general setting of fractional tournaments, is now immediate. We just sort the vertices according to their indegrees ω^-. By Lemma 10, the obtained ordering is minimum, hence it is both ctw-optimal and OLA-optimal.

Lemma 10 cannot be generalized to the semi-complete setting, as there are semi-complete digraphs that do not admit any minimum ordering.

We now give a 2-approximation algorithm for general semi-complete digraphs. The main idea is to relax a given semi-complete digraph to a fractional tournament. 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

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

In particular we have $\mathsf{ctw}(D)/2 \leq \mathsf{ctw}(T_D) \leq \mathsf{ctw}(D)$ and $\mathrm{OLA}(D)/2 \leq \mathrm{OLA}(T_D) \leq \mathrm{OLA}(D)$. The proof of Theorem 8 is now immediate: just output any sorted ordering of T_D.

3 Turing kernel

In this section we prove Theorem 4, that is, we give a quadratic Turing kernel for the problem of computing the cutwidth of a semi-complete digraph. The essence of our approach is encapsulated in the following lemma. Intuitively, it 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 11.** *Let $D = (V, E)$ be a semi-complete digraph. Let π and σ be two vertex orderings of D such that $\mathsf{ctw}(D, \sigma) \leq \mathsf{ctw}(D, \pi) = c$. Suppose further that $m \in [4c, |V| - 4c]$ is such that in D there is a family of $|E_\pi^m|$ arc-disjoint paths leading from $\pi_{>m+4c}$ to $\pi_{\leq m-4c}$. Then there exists a vertex ordering σ^* such that:*

- $\sigma_{\leq m}^* = \pi_{\leq m}$;
- *for every j with $j \leq m - 4c$ or $j > m + 4c$, we have $\sigma_j^* = \sigma_j$;*
- $\mathsf{ctw}(D, \sigma^*) \leq \mathsf{ctw}(D, \sigma)$.

The intuition behind Lemma 11 is as follows. Consider σ^* as rearranged σ. The second condition says that this rearrangement is local: it affects only vertices at positions in the range $[m - 4c + 1, m + 4c]$. 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, 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.

Proof of Lemma 11. We first establish the following basic observation on the relation between orderings π and σ.

▶ **Claim 12 (♠).** *In the ordering σ, every vertex of $\pi_{\leq m-4c}$ is placed before every vertex of $\pi_{>m}$, and every vertex of $\pi_{\leq m}$ is placed before every vertex of $\pi_{>m+4c}$.*

The proof of the claim naturally follows by finding, say for each $u \in \pi_{\leq m-4c}$ and $v \in \pi_{>m}$, sufficiently many vertices that are both outneighbors of u and inneighbors of v.

Let $\sigma_{\leq}$ and $\sigma_{>}$ denote the restriction of σ to $\pi_{\leq m}$ and $\pi_{>m}$, respectively. Then, define σ^* to be the concatenation of $\sigma_{\leq}$ and $\sigma_{>}$. By the construction we have $\pi_{\leq m} = \sigma_{\leq m}^*$, so the first condition is satisfied. For the second condition, observe that by Claim 12, every vertex of $\pi_{\leq m-4c}$ 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 $m - 4c$ 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 $j \leq m - 4c$. A symmetric argument shows that $\sigma_j^* = \sigma_j$ also for all $j > m + 4c$.

It remains to prove that $\mathsf{ctw}(D, \sigma^*) \leq \mathsf{ctw}(D, \sigma)$. Consider any $j \in [0, |V|]$; we need to prove that $|E_{\sigma^*}^j| \leq \mathsf{ctw}(D, \sigma)$. By the second condition we have that $E_{\sigma^*}^j = E_\sigma^j$ when $j \leq m - 4c$ or $j \geq m + 4c$, and $|E_\sigma^j| \leq \mathsf{ctw}(D, \sigma)$ by definition. Hence, we are left with checking the inequality for j satisfying $m - 4c < j < m + 4c$.

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 $\delta(\sigma_{\leq j}^*) \leq \mathsf{ctw}(D, \sigma)$.

Let x be the vertex at position j in σ^* and let X be the set of all vertices placed not after x in σ, including x itself. 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

$$\delta(\sigma^*_{\leq j}) = \delta(X \cap \pi_{\leq m}) \leq \delta(X) + \delta(\pi_{\leq m}) - \delta(X \cup \pi_{\leq m}). \tag{1}$$

As X is a prefix of σ by definition, we have $\delta(X) \leq \mathsf{ctw}(D, \sigma)$. Hence, by (1), in order to prove that $\delta(\sigma^*_{\leq j}) \leq \mathsf{ctw}(D, \sigma)$, it suffices to prove that $\delta(X \cup \pi_{\leq m}) \geq \delta(\pi_{\leq m})$.

Denote $d = \delta(\pi_{\leq m}) = |E^m_\pi|$ and recall that there is a family of d arc-disjoint paths leading from $\pi_{>m+4c}$ to $\pi_{\leq m-4c}$. In particular, this means that for each set A with $A \supseteq \pi_{\leq m-4c}$ and $A \cap \pi_{>m+4c} = \emptyset$, each of these paths has to contribute to $\delta(A)$, implying $\delta(A) \geq d$.

Therefore, it suffices to show that $X \cup \pi_{\leq m} \supseteq \pi_{\leq m-4c}$ and $(X \cup \pi_{\leq m}) \cap \pi_{>m+4c} = \emptyset$. While the first assertion is trivial, the second is equivalent to $X \cap \pi_{>m+4c} = \emptyset$. For this, observe that by definition all elements of X are placed not after x in σ, and x belongs to $\pi_{\leq m}$. However, by Claim 12 all vertices of $\pi_{>m+4c}$ are placed in σ after all vertices of $\pi_{\leq m}$, in particular after x. This implies that X and $\pi_{>m+4c}$ are disjoint, which proves that $\delta(X \cup \pi_{\leq m}) \geq d$ and, consequently as discussed above, also that $\delta(\sigma^*_{\leq j}) \leq \mathsf{ctw}(D, \sigma)$.

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 (1), we are left with proving that $\delta(X \cap \pi_{\leq m}) \geq \delta(\pi_{\leq m})$, which follows by a symmetric reasoning.   ◀

Our goal now is to construct an approximate ordering π where we will be able to find many positions m to which Lemma 11 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 13.** 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 $\min_{a \leq i \leq b} |E^i_\pi|$.

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 an arc 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. [20]. An essentially equivalent notion of *linked orderings* was used by Chudnovsky and Seymour [4] in the context of immersions in tournaments. Also, Giannopoulou et al. [12] used this concept to study immersion obstructions for the cutwidth of undirected graphs. A careful analysis of the arguments of [4, 12] yields the following.

▶ **Lemma 14** ([4, 12]). *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 $\mathsf{ctw}(D, \pi^*) \leq \mathsf{ctw}(D, \pi)$.*

Next, we introduce the concept of a *milestone*. Intuitively, a milestone is a position where Lemma 11 can be applied, provided the ordering is lean.

▶ **Definition 15.** 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 $|E^m_\pi| \leq |E^i_\pi|$ 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 $\min_{m-\alpha \leq i \leq m+\alpha} |E^i_\pi| = |E^m_\pi|$, 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 $4c$ satisfies the prerequisite of Lemma 11 about the existence of arc-disjoint paths. We now observe that in an ordering of small width milestones occur often.

▶ **Lemma 16 (♠).** *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.*

Having gathered all the tools, we are finally ready to prove Theorem 4.

Proof of Theorem 4. By Theorem 8, we can compute in polynomial time a vertex ordering π_0 of D such that $\mathsf{ctw}(D, \pi_0) \leq 2 \cdot \mathsf{ctw}(D)$. If $\mathsf{ctw}(D, \pi_0) > 2c$, we can conclude that $\mathsf{ctw}(D) > c$ and report this answer, so let us assume that $\mathsf{ctw}(D, \pi_0) \leq 2c$. By applying the algorithm of Lemma 14 to π_0, we can compute in polynomial time a lean ordering π such that $\mathsf{ctw}(D, \pi) \leq \mathsf{ctw}(D, \pi_0) \leq 2c$. In the following we assume w.l.o.g. that $|V| > 16c$, 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 $16c$. Observe that 0 and $|V|$ are always π-milestones, and they differ by more than $16c$. Starting from the set $\{0, |V|\}$, we compute an inclusion-wise maximal dispersed set $0 = m_0 < m_1 < m_2 < \ldots < m_\ell = |V|$ of π-milestones of span $8c$. More precisely, whenever some π-milestone of span $8c$ 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 $m_{i+1} - m_i \leq 32c^2 + 32c + 1$ for each $i \in [1, \ell - 1]$, for otherwise the range $[m_i + 16c + 1, m_{i+1} - 16c - 1]$ would contain more than $32c^2$ vertices, so by Lemma 16 we would be able to find in it a π-milestone of span $8c$ that could be added to the constructed dispersed set.

Thus, π is partitioned into ℓ blocks $B_1, \ldots, B_\ell$, each of length at most $32c^2 + 32c + 1$, such that the j-th block B_j is equal to $\{\pi_{m_{j-1}+1}, \pi_{m_{j-1}+2}, \ldots, \pi_{m_j}\}$. For each $j \in [1, \ell]$, let A_j be defined as B_j augmented with the following vertices:

- vertices at positions in ranges $[\max(1, m_{j-1} - 8c + 1), m_{j-1}]$ and $[m_j + 1, \min(|V|, m_j + 8c)]$,
- all heads of arcs from $E_\pi^{m_{j-1}-8c}$, and all tails of arcs from $E_\pi^{m_j+8c}$.

Since the width of π is at most $2c$, we have that $|A_j| \leq |B_j| + 20c = \mathcal{O}(c^2)$.

For $j \in [1, \ell]$, let us denote $D_j = D[A_j]$. To prove the theorem, it now suffices to show that $\mathsf{ctw}(D) \leq c$ if and only if $\mathsf{ctw}(D_j) \leq c$ 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 $\mathsf{ctw}(D_j) \leq c$ for each $j \in [1, \ell]$, then $\mathsf{ctw}(D) \leq c$.

Take any $j \in [1, \ell - 1]$. As m_j is a π-milestone of span $8c$, we have $\min_{m_j - 8c \leq i \leq m_j + 8c} |E_\pi^i| = |E_\pi^{m_j}|$. Since π is lean, there is a family $\mathcal{F}_j$ of $|E_\pi^{m_j}|$ arc-disjoint paths in D leading from $\pi_{>m_j+8c}$ to $\pi_{\leq m_j-8c}$. We can assume w.l.o.g. that each internal (non-endpoint) vertex of each of these paths has position between $m_j + 8c + 1$ and $m_j - 8c$ in π. Hence, in particular, each path of $\mathcal{F}_j$ starts with an arc of $E_\pi^{m_j+8c}$ and ends with an arc of $E_\pi^{m_j-8c}$. 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 $m_{j-1} - 8c + 1$ and $m_{j-1} + 8c$ in π are included in the vertex set of D_j, they are at positions between $m' - 8c + 1$ and $m' + 8c$ in π', and hence the paths of $\mathcal{F}_{j-1}$ in D_j lead from $\pi'_{>m'+8c}$ to $\pi'_{\leq m'-8c}$. 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 11 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 11, σ^* differs from σ by a rearrangement of vertices at positions between $m' - 8c + 1$ and $m' + 8c$.

Now we define m'' to be the position of π_{m_j} in π', so that $\pi'_{\le m''} = \pi_{\le m_j} \cap V(D_j)$. A symmetric reasoning, which uses the fact that $\mathcal{F}_j$ is also entirely contained in D_j, shows that Lemma 11 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 $\sigma^{**}_{\le m''} = \pi'_{\le m''} = \pi_{\le m_j} \cap V(D_j)$. Again, by Lemma 11 we have that σ^* and σ^{**} differ by a rearrangement of vertices at positions $m'' - 8c + 1$ and $m'' + 8c$. Since $m_j - m_{j-1} > 16c$ by construction, we infer that this rearrangement does not change the prefix of length m', and hence we still have $\sigma^{**}_{\le m'} = \pi'_{\le m'} = \pi_{\le m_{j-1}} \cap V(D_j)$. 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 11 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 $V(D_j) \cap \pi_{\le m_{j-1}}$ and vertices to the right of this infix are the vertices of $V(D_j) \cap \pi_{>m_j}$. Define an ordering σ of D by first restricting every ordering σ^j to B_j, and then concatenating all the obtained orderings for $j = 1, 2, \ldots, \ell$. Since we assumed that $\mathsf{ctw}(D_j) \le c$ for each $j \in [1, \ell]$, and each ordering σ^j is ctw-optimal on D_j, we have that $\mathsf{ctw}(D_j, \sigma^j) \le c$ 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 $\mathsf{cuts}\langle D^j, \sigma^j \rangle$ corresponding to the vertices of B_j is equal to the infix of the cutvector $\mathsf{cuts}\langle D, \sigma \rangle$ corresponding to the vertices of B_j. This shows that

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

hence we are done.   ◄

Regarding the bounds on sizes of c-cutwidth-minimal semi-complete digraphs (Theorems 5 and 6), we will give a full exposition in the complete version of the paper. Essentially, Theorem 5 follows easily by considering applying the algorithm of Theorem 4 on a c-cutwidth-minimal semi-complete digraph for parameter $c - 1$, while for Theorem 6 we need to use the understanding of minimum orderings in tournaments in the spirit of Lemma 10.

4   Lower bounds

In this section, we prove Theorem 1, which provides almost tight lower bounds for the complexity of computing the cutwidth and the Ola-cost of a semi-complete digraph. 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 9.

Let us introduce some notation. For a formula φ in CNF, the variable and clause sets of φ are denoted by $\mathrm{vars}(\varphi)$ and $\mathrm{cls}(\varphi)$, respectively. A variable assignment $\alpha \colon \mathrm{vars}(\varphi) \to \{\bot, \top\}$ *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 tuple of size $14m + 1$ such that:

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

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

▶ **Lemma 17.** *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 $\mathsf{cuts}\langle D(\varphi), \pi \rangle \preceq \lambda_m$;*
2. *if φ is NAE-satisfiable, then there exists a vertex ordering π with $\mathsf{cuts}\langle D(\varphi), \pi \rangle = \lambda_m$;*
3. *if there is a vertex ordering π with $\max\{\mathsf{cuts}\langle D(\varphi), \pi \rangle\} \geq 11m$, then φ is NAE-satisfiable.*

Note that Lemma 17 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* $11m$. The main idea is that this maximization will be later turned into minimization by complementing the digraph, which also yields a semi-complete digraph since $D(\varphi)$ is basic.

Proof of Lemma 17. 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 $\mathrm{vars}(\varphi)$ 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 $x \in \mathrm{vars}(\varphi)$, let p_x be the number of occurrences of x in the clauses of φ; hence $3m = \sum\limits_{x \in \mathrm{vars}(\varphi)} p_x$ and $p_x \geq 2$ for each $x \in \mathrm{vars}(\varphi)$. We finally assume the clauses and literals are ordered, so we may say that a literal ℓ_x is the i_xth occurrence of variable x in the clauses of φ, with $i_x \in [1, p_x]$.

We now describe the construction of $D(\varphi)$. For every variable $x \in \mathrm{vars}(\varphi)$ construct a *variable gadget* G_x, which is a directed cycle of length $2p_x$ with vertices named as follows:
$$\bot_1^x \to \top_1^x \to \bot_2^x \to \top_2^x \to \ldots \to \bot_{p_x}^x \to \top_{p_x}^x \to \bot_1^x.$$
Note that this cycle has no symmetric arcs since $p_x > 1$.

Then, for every clause $C \in \mathrm{cls}(\varphi)$, where $C = \ell_x \vee \ell_y \vee \ell_z$ for literals of variables $x, y, z \in \mathrm{vars}(\varphi)$, 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 $(\top_{\ell_x}^C, \top_{\ell_y}^C), (\top_{\ell_y}^C, \top_{\ell_z}^C), (\top_{\ell_z}^C, \top_{\ell_x}^C)$.
- The arcs $(\top^C, \top_{\ell_x}^C), (\top^C, \top_{\ell_y}^C)$ and $(\top^C, \top_{\ell_z}^C)$ from $\top^C$ to the vertices of V_T^C.

Similarly, construct the $\bot$-*clause gadget* $G_\bot^C$, which is isomorphic to $G_\top^C$, but with vertices named $\bot$. Gadgets $G_\top^C$ and $G_\bot^C$ 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 in the second. We use the gadget $G_\top^C$ to verify that α satisfies C, whereas the gadget $G_\bot^C$ 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_xth occurrence of x. If $\ell_x = x$ then add two arcs $(\top_{\ell_x}^C, \bot_{i_x}^x)$ and $(\bot_{\ell_x}^C, \top_{i_x}^x)$, and if $\ell_x = \neg x$ then add two arcs $(\top_{\ell_x}^C, \top_{i_x}^x)$ and $(\bot_{\ell_x}^C, \bot_{i_x}^x)$.

This concludes the construction of $D(\varphi)$. Clearly $D(\varphi)$ is basic, and a straightforward verification using the equality $3m = \sum\limits_{x \in \mathrm{vars}(\varphi)} p_x$ shows that conditions $|V(D(\varphi))| = 14m$ and $|E(D(\varphi))| = 24m$ hold as well. The complete proof of the three lemma statements will appear in the full version of the paper. In order to show the main gist of the reduction, we sketch now the proof of the third claim.

We prove the following statement: for any vertex subset $A \subseteq V$, it always holds that $|E(A, V \setminus A)| \leq 11m$. Note that this in particular implies that the cutwidth of any ordering of $D(\varphi)$ is at most $11m$, which is a part of the verification of the first claim. Denote $F = E(A, V \setminus A)$. First, consider any variable $x \in \mathrm{vars}(\varphi)$. 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 $C = \ell_x \vee \ell_y \vee \ell_z \in \mathrm{cls}(\varphi)$. Let $R_\top^C$ be the set of three arcs connecting G^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 $|E(A, V \setminus A)|$. We now distinguish cases depending on the size 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 $\bot$-clause gadgets, yielding $|F \cap (E(G_\bot^C) \cup R_\bot^C)| \leq 4$, where $R_\bot^C$ is defined analogously. Since the sets $E(G_x)$ for $x \in \mathrm{vars}(\varphi)$ and $E(G_\top^C) \cup R_\top^C \cup E(G_\bot^C) \cup R_\bot^C$ for $C \in \mathrm{cls}(\varphi)$ form a partition of $E(D(\varphi))$, we immediately get that $|F| \leq \sum_{x \in \mathrm{vars}(\varphi)} p_x + 8|\mathrm{cls}(\varphi)| = 11m$.

To verify the third claim of the lemma, note that if there is some vertex ordering of cutwidth at least $11m$, then there is some set A with $|E(A, V \setminus A)| \geq 11m$. Hence, for such all the inequalities used above are in fact equalities. In particular, in every variable gadget G_x, the vertices belong to A and to $V(D) \setminus A$ alternately. This gives us two possibilities for every variable gadget, which naturally defines a variable assignment α for the formula φ. The fact that we have equalities also in each clause gadget $G_\top^C$ and $G_\bot^C$ ensures that each clause C is satisfied both by α and $\neg\alpha$. Hence α NAE-satisfies φ. ◀

We now proceed to complementing the obtained digraph. 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 17 are transformed under complementation yields the following.

▶ **Lemma 18 (♠).** *The complement of $D(\varphi)$ is a semi-complete digraph $\bar{D}(\varphi)$ satisfying:*
1. *for every vertex ordering π, we have $\bar{\lambda}_m \preceq \mathsf{cuts}\langle\bar{D}(\varphi), \pi\rangle$;*
2. *if φ is NAE-satisfiable, then there exists a vertex ordering π with $\mathsf{cuts}\langle\bar{D}(\varphi), \pi\rangle = \bar{\lambda}_m$;*
3. *if $\bar{D}(\varphi)$ admits a vertex ordering π of width at most $49m^2 - 11m$, then φ is NAE-satisfiable.*

Thus, Lemma 18 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 18, 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 1 essentially complete.

The reduction of Lemma 17 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 problem is polynomial-time solvable, which suggests looking at the parameterization by the number of vertices incident to symmetric arcs. 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.

▶ **Theorem 19 (♠).** *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.*

References

1 Jørgen Bang-Jensen and Gregory Gutin. *Digraphs – theory, algorithms and applications.* Springer, 2002.

2 Daniel Binkele-Raible, Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. *ACM Transactions on Algorithms*, 8(4):38, 2012.

3 Maria Chudnovsky, Alexandra Fradkin, and Paul Seymour. Tournament immersion and cutwidth. *Journal of Combinatorial Theory, Series B*, 102(1):93–101, 2012.

4 Maria Chudnovsky and Paul Seymour. A well-quasi-order for tournaments. *Journal of Combinatorial Theory, Series B*, 101(1):47–53, 2011.

5 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015.

6 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

7 Andrew Drucker. New limits to classical and quantum instance compression. *SIAM journal of Computing*, 44(5):1443–1479, 2015.

8 Fedor V. Fomin and Michał 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.

9 Alexandra Fradkin. *Forbidden structures and algorithms in graphs and digraphs.* PhD thesis, Princeton University, 2011.

10 Alexandra Ovetsky Fradkin and Paul D. Seymour. Edge-disjoint paths in digraphs with bounded independence number. *Journal of Combinatorial Theory, Series B*, 110:19–46, 2015.

11 Valentin Garnero and Mathias Weller. Parameterized certificate dispersal and its variants. *Theoretical Computer Science*, 622:66–78, 2016.

12 Archontia C. Giannopoulou, Michał Pilipczuk, Jean-Florent Raymond, Dimitrios M. Thilikos, and Marcin Wrochna. Cutwidth: obstructions and algorithmic aspects. *CoRR*, abs/1606.05975, 2016. To appear in Proceedings of IPEC 2016.

13 Danny Hermelin, Stefan Kratsch, Karolina Sołtys, Magnus Wahlström, and Xi Wu. A completeness theory for polynomial (Turing) kernelization. *Algorithmica*, 71(3):702–730, 2015.

14 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

15 Bart 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.

16 Michał 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.

17 Michał Pilipczuk. *Tournaments and optimality: new results in parameterized complexity.* PhD thesis, University of Bergen, Norway, 2013.

18 Alexander Schäfer, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.

19 Thomas J. Shaefer. The complexity of satisfiability problems. In *STOC 1978*, pages 216–226. ACM, 1978.

20 Robin Thomas. A menger-like property of tree-width: the finite case. *Journal of Combinatorial Theory Series B*, 48(1):67–76, 1990.

21 Stéphan Thomassé, Nicolas Trotignon, and Kristina Vušković. A polynomial Turing-kernel for Weighted Independent Set in bull-free graphs. *Algorithmica*, 77(3):619–641, 2017.