# Computing cutwidth and pathwidth of semi-complete digraphs via degree orderings

**Michał Pilipczuk**[*1]

1 **Department of Informatics, University of Bergen, Norway**
michal.pilipczuk@ii.uib.no

## Abstract

The notions of cutwidth and pathwidth of digraphs play a central role in the containment theory for tournaments, or more generally semi-complete digraphs, developed in a recent series of papers by Chudnovsky, Fradkin, Kim, Scott, and Seymour [2, 3, 4, 7, 8, 10]. In this work we introduce a new approach to computing these width measures on semi-complete digraphs, via degree orderings. Using the new technique we are able to reprove the main results of [2, 8] in a unified and significantly simplified way, as well as obtain new results. First, we present polynomial-time approximation algorithms for both cutwidth and pathwidth, faster and simpler than the previously known ones; the most significant improvement is in case of pathwidth, where instead of previously known $O(OPT)$-approximation in fixed-parameter tractable time [5] we obtain a constant-factor approximation in polynomial time. Secondly, by exploiting the new set of obstacles for cutwidth and pathwidth, we show that topological containment and immersion in semi-complete digraphs can be tested in single-exponential fixed-parameter tractable time. Finally, we present how the new approach can be used to obtain exact fixed-parameter tractable algorithms for cutwidth and pathwidth, with single-exponential running time dependency on the optimal width.

**1998 ACM Subject Classification** G.2.2 Graph Algorithms

**Keywords and phrases** semi-complete digraph, tournament, pathwidth, cutwidth

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2013.197

## 1 Introduction

**Minors of (di)graphs.** The *Graph Minors* series of Robertson and Seymour not only resolved the Wagner's Conjecture, but also provided a number of algorithmic tools for investigating topological structure of graphs. A useful concept that is repeatedly used in many arguments in this theory, is the WIN/WIN approach that can be formulated as follows. We introduce a graph width measure $\mu$, usually defined as the optimal width of some decomposition. The crucial part is to prove a structural theorem that asserts that graphs with large measure $\mu$ contain certain obstacles to admitting a simpler decomposition. These obstacles can be exploited in many ways: either we can immediately provide an answer to the algorithmic problem under consideration, or, for instance, find a vertex in the obstacle whose deletion does not change the answer. If no obstacle is present, we can find a decomposition of small width and try to apply algorithms based on the dynamic programming paradigm. The most classical measure $\mu$ is the treewidth of the graph, accompanied by a grid minor as the obstacle; however, the framework has found applications in many other contexts as well.

Since the beginning of the graph minors project there was a thrilling question, to what extent similar results can be obtained in the directed setting. The answer is still unclear;

* Supported by the ERC grant "Rigorous Theory of Preprocessing", reference 267959.


30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 197–208
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

however, it seems that there is no hope for a positive outcome in the general case. There are several reasons for this. Despite numerous attempts, we still lack any good width measure that would be an analogue of treewidth [9]; moreover, most of the containment problems are already NP-hard for small, constant-size graphs $H$ that are to be embedded into input graph $G$ [6]. Therefore, the scope of research moved to identifying subclasses of digraphs where construction of a sound containment theory could be possible. We invite an interested reader to the introduction of a recent work of Fomin and the current author [5] for a more detailed overview of the status of containment problems in digraphs.

**Tournaments.** In a recent series of papers, Chudnovsky, Fradkin, Kim, Scott, and Seymour [2, 3, 4, 7, 8, 10] identify tournaments as a class where an elegant containment theory can be constructed. Recall that a tournament is a digraph where every two vertices are connected by an arc directed in one of two possible directions; the results hold also for a slightly more general class of *semi-complete* digraphs, where we additionally allow existence of two arcs directed in opposite directions. The central notions of the theory are two width measures of digraphs: *cutwidth* and *pathwidth*. The first one is based on the ordering of vertices and resembles classical cutwidth in the undirected setting [12], with the exception that only arcs directed forward in the ordering contribute to the width of a cut. The second one is a similar generalization of undirected pathwidth. See Section 2.2 for formal definitions.

Chudnovsky, Fradkin, and Seymour [2] prove a structural theorem that provides a set of obstacles for admitting an ordering of small (cut)width; a similar theorem for pathwidth was proven by Fradkin and Seymour [8]. A large enough obstacle for cutwidth admits every fixed-size digraph as an immersion, and the corresponding is true also for pathwidth and topological containment. Basing on the first result, Chudnovsky and Seymour [4] were able to show that immersion is a well-quasi-order on the class of semi-complete digraphs. The same is not true for topological containment, but recently Kim and Seymour [10] introduced a relaxed notion of *minor* order, which indeed is a well-quasi-order of semi-complete digraphs.

As far as the algorithmic aspects of the work of Chudnovsky, Fradkin, Kim, Scott, and Seymour are concerned, the original proofs of the structural theorems can be turned into approximation algorithms, which given a semi-complete digraph $T$ and an integer $k$ find a decomposition of $T$ of width $O(k^2)$, or provide an obstacle for admitting decomposition of width at most $k$. For cutwidth the running time is polynomial, but for pathwidth it is $O(|V(T)|^{m(k)})$ for some function $m$; this excludes usage of this approximation as a subroutine in any FPT algorithm[1], for instance, for topological containment testing. This gap has been bridged by Fomin and the current author [5] by presenting an FPT approximation algorithm for pathwidth that again either finds a path decomposition of $T$ of width $O(k^2)$ or provides an obstacle for admitting path decomposition of width at most $k$, but runs in $O(2^{O(k \log k)}|V(T)|^3 \log |V(T)|)$ time. The approach in [5] is based on replacing the crucial part of the algorithm of Fradkin and Seymour that was implemented by a brute-force enumeration, by a more refined argument based on the colour coding technique.

As far as computation of optimal decompositions is concerned, by the well-quasi-ordering result of Chudnovsky and Seymour [4] we have that the class of semi-complete digraphs of cutwidth bounded by a constant is characterized by a finite set of forbidden immersions; the result of Kim and Seymour [10] proves that we can infer the same conclusion about pathwidth and minors. Having approximated the corresponding parameter, in FPT time we can check if any of these forbidden structures is contained in a given semi-complete digraph,

[1] An algorithm for a parameterized problem is said to be *fixed-parameter tractable* (FPT), if it runs in time $f(k) \cdot n^c$ on instances of size $n$ and parameter $k$, for some function $f$ and constant $c$.

**Table 1** Comparison of previously known algorithms and the results of this work. The algorithms for cutwidth and pathwidth take as input a semi-complete digraph on $n$ vertices and an integer $k$. The approximation algorithms can output a decomposition of larger width (guarantees are in the corresponding cells) or conclude that a decomposition of width at most $k$ does not exist. The exact algorithms either construct a decomposition of width at most $k$ or conclude that this is impossible.

| Problem | Previous results | This work |
|---|---|---|
| Cutwidth approximation | $O(n^3)$ time, width $O(k^2)$ [2] | $O(n^2)$ time, width $O(k^2)$ |
| Cutwidth exact | $O(f(k) \cdot n^3)$ time, non-uniform, non-constructive [2, 4] | $O(2^{O(k)} \cdot n^2)$ time |
| Pathwidth approximation | $O(2^{O(k \log k)} \cdot n^3 \log n)$ time, width $O(k^2)$ [5] | $O(kn^2)$ time, width $7k$ |
| Pathwidth exact | $O(f(k) \cdot n^3 \log n)$ time, non-uniform, non-constructive [5, 10] | $O(2^{O(k \log k)} \cdot n^2)$ time |
| Immersion | $O(f(\|H\|) \cdot n^3)$ time [2] | $O(2^{O(\|H\|^2 \log \|H\|)} \cdot n^2)$ time |
| Topological containment | $O(f(\|H\|) \cdot n^3 \log n)$ time [5] | $O(2^{O(\|H\| \log \|H\|)} \cdot n^2)$ time |

using dynamic programming. This gives FPT exact algorithms computing cutwidth and pathwidth; however, they are both non-uniform — the algorithms depend on the set of forbidden structures which is unknown — and non-constructive — they provide just the value of the width measure, and not the optimal decomposition. To the best of author's knowledge, nothing better was known for these problems.

**Our results and techniques.** In this paper we present a new approach to computing cutwidth and pathwidth of semi-complete digraphs, via degree orderings. Using the new technique we are able to reprove the structural theorems for both parameters in a unified and simplified way, obtaining in both cases polynomial-time algorithms and, in case of pathwidth, a constant-factor approximation. The technique can be also used to develop FPT exact algorithm for both width measures with single-exponential dependency of the running time on the optimal width, as well as to trim the dependency of the running time on the size of the tested digraph in the topological containment and immersion tests to single-exponential. All the algorithms presented in this paper have quadratic dependency on the number of vertices of a given semi-complete digraph; note that this is *linear* in the input size.

The crucial observation of the new approach can be intuitively formulated as follows. Consider a semi-complete digraph with a large number of vertices, but admitting a path decomposition of very small width. Take any vertex $v$ that appears in a small number of bags. Observe that for each vertex $w$ that is located much further in the decomposition, the arc between $v$ and $w$ must be directed from $w$ to $v$. Similarly, if $w$ is much earlier, then the arc must be directed from $v$ to $w$. Hence, the outdegree of $v$ is more or less equal to the number of vertices that appear before $v$ in the decomposition; the only aberrations that can occur are due to vertices that interact with $v$ within the bags, but their number is limited by the width of the decomposition. Therefore, intuitively any ordering of the vertices with respect to increasing outdegrees should be a good approximation of the order in which the vertices appear in the optimal path decomposition.

In fact, for cutwidth this is true even in formal sense. We prove that any outdegree ordering of vertices of a semi-complete digraph $T$ has width at most $O(\mathbf{ctw}(T)^2)$, hence we have a trivial approximation algorithm that sorts the vertices with respect to outdegrees.

The exact algorithm for cutwidth is based on an observation that an optimal ordering and any outdegree ordering can differ only locally: if $X$ is the set of $\ell$ first vertices in the optimal ordering, then $X$ contains first $\ell - O(k)$ vertices of the degree ordering, and is contained in first $\ell + O(k)$ vertices of the outdegree ordering. Hence, we scan through the outdegree ordering with a dynamic program, maintaining a bit mask of length $O(k)$ denoting which vertices of an appropriate interval are contained in constructed prefix of an optimal ordering.

The case of pathwidth, which is of our main interest, is more complicated. We formalize the intuitive outdegree ordering argument as follows: we prove that existence of $5k + 2$ vertices with outdegrees mutually not differing by more than $k$ already forms an obstacle for admitting a path decomposition of width at most $k$; we call this obstacle a *degree tangle*. Hence, any outdegree ordering of vertices of a given semi-complete digraph $T$ of small pathwidth must be already quite spread: it does not contain larger clusters of vertices with similar outdegrees. This spread argument is crucial in all our reasonings, and shows that finding new structural obstacles can significantly simplify our view of the problem.

Both the approximation and exact algorithm for pathwidth use the concept of scanning through the outdegree ordering with a *window* — an interval in the ordering containing $5k$ vertices. By the outdegree spread argument, at each point we know that the vertices on the left side of the window have outdegrees smaller by more than $k$ that the ones on the right side; otherwise we would have a degree tangle. For approximation, we construct the consecutive bags by greedily taking the window and augmenting this choice with a small coverage of arcs jumping over it. The big gap between outdegrees on the left and on the right side of the window ensures that nonexistence of a small coverage is also an evidence for not admitting a decomposition of small width. For exact algorithm, we identify a set of $O(k^2)$ vertices around the window, about which we can safely assume that the bag is contained in it. Then we run a similar dynamic programming algorithm as in case of cutwidth.

The most technical part in the approximation and exact algorithms for pathwidth is the choice of vertices outside the window to cover the arcs jumping over it. It turns out that this problem can be expressed as trying to find a small vertex cover in an auxiliary bipartite graph. However, in order to obtain a feasible path decomposition we cannot choose the vertex cover arbitrarily — it must behave consistently as the window slides through the ordering. To this end, in the approximation algorithm we use a 2-approximation of the vertex cover based on the matching theory. In the exact algorithm we need more restrictions, as we seek a subset that contains *every* sensible choice of the bag. Therefore, we use an $O(OPT)$-approximation of vertex cover based on the classical Buss kernelization routine for the problem [1], which enables us to use stronger arguments to reason which vertices can be excluded from consideration.

Finally, we observe that the obtained set of obstacles for pathwidth have much better properties than the ones introduced by Fradkin and Seymour [8]. We show that there is a constant multiplicative factor relation between the sizes of obstacles found by the algorithm and the minimum size of a digraph that cannot be embedded into them. Hence, in order to test if $H$ is topologically contained in a given semi-complete digraph $T$, we just need to run the pathwidth approximation with parameter $O(|H|)$, and in case of finding an obstacle just provide a positive answer. This trims the dependency of running time of the topological subgraph and immersion tests to single exponential in terms of the size of tested subgraph, compared to multiple-exponential as presented in [5, 2].

**Organization.** In Section 2 we introduce notation and main definitions. In Section 3 we present the obstacles and prove their basic properties. In Section 4 we deal with cutwidth and in Section 5 with pathwidth. In Section 6 we apply the introduced tools to the containment

problems, while Section 7 gathers concluding remarks. The full version of this paper is available online [11], and contains all the proofs removed from this extended abstract due to space constraints (denoted by ♠). In this extended abstract we try to explain in details the approximation algorithm for pathwidth, which is the main result of the paper.

# 2 Preliminaries

## 2.1 Notation and basic definitions

We use standard graph notation. If $G$ is a (di)graph, by $V(G)$ we denote the vertex set of $G$ and by $E(G)$ the edge (arc) set of $G$; we define size of $G$ to be equal to $|G| = |V(G)| + |E(G)|$. By $G[X]$ we denote the sub(di)graph induced by $X$ and for $v \in V(G)$ by $G \setminus v$ we denote $G[V(G) \setminus \{v\}]$. For $X, Y \subseteq V(G)$ we define $E(X,Y) = \{(v,w) \in E(G) \mid v \in V, w \in W\}$. If $G$ is undirected, then $N(v)$ denotes the set of neighbours of $v$; for directed graphs, by $N^+(v), N^-(v)$ we denote the sets of outneighbours and inneighbours of $v$, respectively. We extend this notion to subsets naturally, e.g., $N(X) = \bigcup_{v \in X} N(v) \setminus X$. We define the *outdegree* and *indegree* of $v$ as $d^+(v) = |N^+(v)|$ and $d^-(v) = |N^-(v)|$, respectively. An *outdegree ordering* is an ordering of vertices with respect to nondecreasing outdegrees.

A digraph is *simple*, if it does not contain loops or multiple arcs; all the digraphs considered in this paper are simple. A digraph is *semi-complete*, if it is simple and for every two distinct vertices $v, w$ **at least** one of the arcs $(v,w)$ or $(w,v)$ is present. Note that we do not exclude existence of both of these arcs, but we exclude existence of, for instance, two arcs $(v,w)$. A semi-complete digraph is a *tournament*, if for every two distinct vertices $v, w$ **exactly** one of the arcs $(v,w)$ or $(w,v)$ is present.

A *separation* in a digraph $T$ is a pair $(A,B)$ such that $A, B \subseteq V(T)$, $A \cup B = V(T)$ and $E(A \setminus B, B \setminus A) = \emptyset$. The *order* of a separation is the size of the *separator*, i.e., $|A \cap B|$.

*Immersion* and *topological containment* are the two main notions of containment that are under consideration in this paper. They are direct analogues of the classical undirected versions; the exact formal definitions can be found in the full version of the paper.

## 2.2 Width parameters of digraphs

▶ **Definition 1.** Given a digraph $G = (V,E)$ and an ordering $\pi$ of $V$, let $\pi[\alpha]$ be the first $\alpha$ vertices in the ordering $\pi$. The *width* of $\pi$ is equal to $\max_{0 \leq \alpha \leq |V|} |E(\pi[\alpha], V \setminus \pi[\alpha])|$; the *cutwidth* of $G$, denoted $\mathbf{ctw}(G)$, is the minimum width over all orderings of $V$.

▶ **Definition 2.** Given a digraph $G = (V,E)$, a sequence $W = (W_1, \ldots, W_r)$ of subsets of $V$ is a *path decomposition of* $G$ if the following conditions are satisfied:

*(i)* $\bigcup_{1 \leq i \leq r} W_i = V$;
*(ii)* $W_i \cap W_k \subseteq W_j$ for $1 \leq i < j < k \leq r$;
*(iii)* $\forall\ (u,v) \in E$, either $u, v \in W_i$ for some $i$ or $u \in W_i$, $v \in W_j$ for some $i > j$.

We call $W_1, \ldots, W_r$ the *bags* of the path decomposition. The *width* of a path decomposition is equal to $\max_{1 \leq i \leq r}(|W_i| - 1)$; the *pathwidth* of $G$, denoted $\mathbf{pw}(G)$, is the minimum width over all path decompositions of $G$.

In [5] it is observed that $\mathbf{pw}(G) \leq 2\mathbf{ctw}(G)$ for every digraph $G$. We say that a path decomposition $W = (W_1, \ldots, W_r)$ is *nice* if it has following two additional properties:

- $W_1 = W_r = \emptyset$;
- for every $i = 1, 2, \ldots, r-1$ we have that $W_{i+1} = W_i \cup \{v\}$ for some vertex $v \notin W_i$, or $W_{i+1} = W_i \setminus \{w\}$ for some vertex $w \in W_i$.

If $W_{i+1} = W_i \cup \{v\}$ then we say that in bag $W_{i+1}$ we *introduce* vertex $v$, while if $W_{i+1} = W_i \setminus \{w\}$ then we say that in bag $W_{i+1}$ we *forget* vertex $w$. Given any path decomposition $W$ of width $p$, in $O(p|V(T)|)$ time we can construct a nice path decomposition $W'$ of the same width in a standard manner: we first introduce empty bags at the beginning and at the end, and then insert new bags between any two consecutive ones by firstly forgetting all the vertices that do not appear in the second bag, and then introducing all the new ones.

Any path decomposition naturally corresponds to a monotonic sequence of separations.

▶ **Definition 3.** A sequence of separations $((A_0, B_0), \ldots, (A_r, B_r))$ is called a *separation chain* if $(A_0, B_0) = (\emptyset, V(T))$, $(A_r, B_r) = (V(T), \emptyset)$ and $A_i \subseteq A_j, B_i \supseteq B_j$ for all $i \leq j$. The *width* of the separation chain is equal to $\max_{0 \leq i \leq r} |A_i \cap B_i|$.

▶ **Lemma 4 (♠).** *The following holds.*

- *Let $W = (W_1, \ldots, W_r)$ be a path decomposition of a digraph $T$ of width at most $p$, where no two consecutive bags are equal. Then sequence $((A_0, B_0), \ldots, (A_r, B_r))$ defined as $(A_i, B_i) = (\bigcup_{j=1}^{i} W_j, \bigcup_{j=i+1}^{r} W_j)$ is a separation chain of width at most $p$.*
- *Let $((A_0, B_0), \ldots, (A_r, B_r))$ be a separation chain in digraph $T$. Then $W = (W_1, \ldots, W_r)$ defined by $W_i = A_i \cap B_{i-1}$ is a path decomposition of width $\max_{1 \leq i \leq r} |A_i \cap B_{i-1}| - 1$.*

Assume that $((A_0, B_0), \ldots, (A_r, B_r))$ is a separation chain corresponding to a nice path decomposition $W$ in the sense of Lemma 4. Since $A_0 = \emptyset$, $A_r = V(G)$, and $|A_i|$ can change by at most 1 between two consecutive separations, for every $\ell$, $0 \leq \ell \leq |V(G)|$, there is some separation $(A_i, B_i)$ for which $|A_i| = \ell$ holds. Let $W[\ell]$ denote any such separation.

## 2.3 Degrees in semi-complete digraphs

Let $T$ be a semi-complete digraph. Note that for every $v \in V(T)$ we have that $d^+(v) + d^-(v) \geq |V(T)| - 1$, and that the equality holds for all $v \in V(T)$ if and only if $T$ is a tournament.

▶ **Lemma 5.** *Let $T$ be a semi-complete digraph. Then the number of vertices of $T$ with outdegrees at most $d$ is at most $2d + 1$.*

**Proof.** Let $A$ be the set of vertices of $T$ with outdegrees at most $d$, and for the sake of contradiction assume that $|A| > 2d + 1$. Consider semi-complete digraph $T[A]$. As in every semi-complete digraph $S$ there is a vertex of outdegree at least $\frac{|V(S)|-1}{2}$, in $T[A]$ there is a vertex with outdegree larger than $d$. As outdegrees in $T[A]$ are not smaller than in $T$, this is a contradiction with the definition of $A$. ◀

▶ **Lemma 6.** *Let $T$ be a semi-complete digraph and let $x, y$ be vertices of $T$ such that $d^+(x) > d^+(y) + \ell$. Then there exist at least $\ell$ vertices that are both outneighbours of $x$ and inneighbours of $y$ and, consequently, $\ell$ vertex-disjoint paths of length 2 from $x$ to $y$.*

**Proof.** Let $\alpha = d^+(y)$. We have that $d^-(y) + d^+(x) \geq |V(T)| - 1 - \alpha + \alpha + \ell + 1 = |V(T)| + \ell$. Hence, by the pigeonhole principle there exist at least $\ell$ vertices of $T$ that are both outneighbours of $x$ and inneighbours of $y$. ◀

# 3 The obstacle zoo

In this section we describe the set of obstacles used by the algorithms. We begin with *short jungles*, enhanced analogues of the $k$-jungle of Fradkin and Seymour [8]. It appears that the enhancement enables us to construct large topological subgraph or immersion models in short jungles in a greedy manner, and this observation is the key to trimming the running

times for topological containment and immersion tests. We continue with further obstacles: degree and matching tangles for pathwidth, and backward tangles for cutwidth, each time proving two lemmas. The first asserts that existence of the structure is indeed an obstacle for having small width, while the second (whose proof is in the full version of the paper) shows that one can constructively find an appropriate short jungle in a sufficiently large obstacle.

## 3.1 Short jungles

▶ **Definition 7.** Let $T$ be a semi-complete digraph and $k, d$ be integers. A $(k,d)$-*short (immersion) jungle* is a set $X \subseteq V(T)$ such that *(i)* $|X| \geq k$; *(ii)* for every $v, w \in X$ there are $k$ vertex-disjoint (edge-disjoint) paths from $v$ to $w$ of length at most $d$.

We remark that in all our algorithms, every short jungle is constructed and stored together with corresponding families of $k$ paths for each pair of vertices.

The restriction on the length of the paths enables us to construct topological subgraph (immersion) models as follows. We choose arbitrary vertices of $X$ as images of vertices and build the images of arcs greedily: if the number of paths between vertices of the short jungle is sufficiently large, then one of them is still untouched by the previously used ones, as each of the previous constructions used at most $d-1$ vertices ($d$ arcs). This provides a linear relation between the obstruction size and the bound on the size of the digraphs that can be realized in the obstruction, assuming that $d$ is a constant.

▶ **Lemma 8 (♠).** *If a digraph $T$ contains a $(dk,d)$-short (immersion) jungle for some $d > 1$, then it admits every digraph $S$ with $|S| \leq k$ as a topological subgraph (as an immersion).*

## 3.2 Degree tangles

▶ **Definition 9.** Let $T$ be a semi-complete digraph and $k, \ell$ be integers. A $(k,\ell)$-*degree tangle* is a set $X \subseteq V(T)$ such that *(i)* $|X| \geq k$; *(ii)* for every $v, w \in X$ we have $|d^+(v) - d^+(w)| \leq \ell$.

▶ **Lemma 10.** *Let $T$ be a semi-complete digraph. If $T$ contains a $(5k+2,k)$-degree tangle $X$, then* $\mathbf{pw}(T) > k$.

**Proof.** For the sake of contradiction, assume that $T$ admits a (nice) path decomposition $W$ of width at most $k$. Let $\alpha = \min_{v \in X} d^+(v)$ and $\beta = \max_{v \in X} d^+(v)$; we know that $\beta - \alpha \leq k$. Let $(A,B) = W[\alpha]$. We know that $|A \cap B| \leq k$ and $|A| = \alpha$.

Firstly, observe that $X \cap (A \setminus B) = \emptyset$. This follows from the fact that vertices in $A \setminus B$ can have outneighbours only in $A$, so their outdegrees are upper bounded by $|A| - 1 = \alpha - 1$. Secondly, $|X \cap (A \cap B)| \leq k$ as $|A \cap B| \leq k$. Thirdly, we claim that $|X \cap (B \setminus A)| \leq 4k+1$. Assume otherwise. By Lemma 5 there exists a vertex $v \in X \cap (B \setminus A)$ whose outdegree in $T[B \setminus A]$ is at least $2k+1$. As $(A,B)$ is a separation and $T$ is semi-complete, all the vertices of $A \setminus B$ are also outneighbours of $v$. Note that $|A \setminus B| = |A| - |A \cap B| \geq \alpha - k$. Hence $v$ has at least $\alpha - k + 2k + 1 = \alpha + k + 1 > \beta$ neighbours, which is a contradiction with $v \in X$.

Summing up the bounds we get $5k+2 \leq |X| \leq k + 4k + 1 = 5k+1$, a contradiction. ◀

▶ **Lemma 11 (♠).** *Let $T$ be a semi-complete digraph and let $X$ be a $(26k,k)$-degree tangle in $T$. Then $X$ contains a $(k,3)$-short jungle, which can be found in $O(k^3|V(T)|^2)$ time.*

## 3.3 Matching tangles

▶ **Definition 12.** Let $T$ be a semi-complete digraph and $k, \ell$ be integers. A $(k,\ell)$-*matching tangle* is a pair of disjoint subsets $X, Y \subseteq V(T)$ such that *(i)* $|X| = |Y| = k$; *(ii)* there exists

a matching from $X$ to $Y$, i.e., there is a bijection $f : X \to Y$ such that $(v, f(v)) \in E(T)$ for all $v \in X$; *(iii)* for every $v \in X$ and $w \in Y$ we have that $d^+(w) > d^+(v) + \ell$.

▶ **Lemma 13.** *Let $T$ be a semi-complete digraph. If $T$ contains a $(k+1, k)$-matching tangle $(X, Y)$, then $\mathbf{pw}(T) > k$.*

**Proof.** For the sake of contradiction assume that $T$ has a (nice) path decomposition $W$ of width at most $k$. Let $\alpha = \min_{w \in Y} d^+(w)$ and let $(A, B) = W[\alpha]$. Recall that $|A \cap B| \leq k$.

Firstly, we claim that $X \subseteq A$. Assume otherwise that there exists some $v \in (B \setminus A) \cap X$. Note that all the vertices of $A \setminus B$ are outneighbours of $v$, so $d^+(v) \geq |A| - k = \alpha - k$. Hence $d^+(v) \geq d^+(w) - k$ for some $w \in Y$, which is a contradiction. Secondly, we claim that $Y \subseteq B$. Assume otherwise that there exists some $w \in (A \setminus B) \cap Y$. Then all the outneighbours of $w$ lie within $A$, so there is less than $\alpha$ of them. This is a contradiction with the definition of $\alpha$.

As $|A \cap B| \leq k$ and there are $k+1$ disjoint pairs of form $(v, f(v)) \in E(T)$ for $v \in X$, we conclude that there must be some $v \in X$ such that $v \in A \setminus B$ and $f(v) \in B \setminus A$. This contradicts the fact that $(A, B)$ is a separation. ◀

▶ **Lemma 14 (♠).** *Let $T$ be a semi-complete digraph and let $(X, Y)$ be a $(5k, 3k)$-matching tangle in $T$. Then $Y$ contains a $(k, 4)$-short jungle, which can be found in $O(k^3|V(T)|)$ time.*

## 3.4 Backward tangles

▶ **Definition 15.** Let $T$ be a semi-complete digraph and $k$ be an integer. A *$k$-backward tangle* is a partition $(X, Y)$ of $V(T)$ such that *(i)* there exist at least $k$ arcs directed from $X$ to $Y$; *(ii)* for every $v \in X$ and $w \in Y$ we have that $d^+(w) \geq d^+(v)$.

▶ **Lemma 16.** *Let $T$ be a semi-complete digraph. If $T$ contains an $(m+1)$-backward tangle $(X, Y)$ for $m = 100k^2 + 22k + 1$, then $\mathbf{ctw}(T) > k$.*

**Proof.** For the sake of contradiction, assume that $V(T)$ admits an ordering $\pi$ of width at most $k$. Let $\alpha$ be the largest index such that $(X_\alpha, Y_\alpha) = (\pi[\alpha], V(T) \setminus \pi[\alpha])$ satisfies $Y \subseteq Y_\alpha$. Similarly, let $\beta$ be the smallest index such that $(X_\beta, Y_\beta) = (\pi[\beta], V(T) \setminus \pi[\beta])$ satisfies $X \subseteq X_\beta$. Note that $|E(X_\alpha, Y_\alpha)|, |E(X_\beta, Y_\beta)| \leq k$. Observe also that $\alpha \leq \beta$; moreover, $\alpha < |V(T)|$ and $\beta > 0$, since $X, Y$ are non-empty.

Let $(X_{\alpha+1}, Y_{\alpha+1}) = (\pi[\alpha+1], V(T) \setminus \pi[\alpha+1])$. By the definition of $\alpha$ there is a unique vertex $w \in X_{\alpha+1} \cap Y$. Take any vertex $v \in V(T)$ and suppose that $d^+(w) > d^+(v) + (k+1)$. By Lemma 6, there exist $k+1$ vertex-disjoint paths of length 2 from $w$ to $v$. If $v$ was in $Y_{\alpha+1}$, then each of these paths would contribute at least one arc to the set $E(X_{\alpha+1}, Y_{\alpha+1})$, contradicting the fact that $|E(X_{\alpha+1}, Y_{\alpha+1})| \leq k$. Hence, every such $v$ belongs to $X_{\alpha+1}$ as well. By Lemma 10 we have that the number of vertices with outdegrees in the interval $[d^+(w) - (k+1), d^+(w)]$ is bounded by $10k+1$, as otherwise they would create a $(10k+2, 2k)$-degree tangle, implying that $\mathbf{pw}(T) > 2k$ and, consequently, $\mathbf{ctw}(T) > k$ (here note that for $k = 0$ the lemma is trivial). As $X_\alpha = X_{\alpha+1} \setminus \{w\}$ is disjoint with $Y$ and all the vertices of $X$ have degrees at most $d^+(w)$, we infer that $|X \setminus X_\alpha| \leq 10k + 1$.

A symmetrical reasoning shows that $|Y \setminus Y_\beta| \leq 10k + 1$. Now observe that

$$\begin{aligned} |E(X, Y)| &\leq |E(X_\alpha, Y)| + |E(X, Y_\beta)| + |E(X \setminus X_\alpha, Y \setminus Y_\beta)| \\ &\leq |E(X_\alpha, Y_\alpha)| + |E(X_\beta, Y_\beta)| + |E(X \setminus X_\alpha, Y \setminus Y_\beta)| \\ &\leq k + k + (10k+1)^2 = 100k^2 + 22k + 1. \end{aligned}$$

This is a contradiction with $(X, Y)$ being an $(m+1)$-backward tangle. ◀

▶ **Lemma 17 (♠).** *Let $T$ be a semi-complete digraph and let $(X, Y)$ be an $m$-backward tangle in $T$ for $m = 109^2k$. Then $X$ or $Y$ contains a $(k, 4)$-short immersion jungle, which can be found in $O(k^3|V(T)|^2)$ time.*

## 4 Algorithms for cutwidth

In this section we present the algorithms for computing cutwidth. We start with the approximation algorithm. The exact algorithm is deferred to the full version of the paper due to space constraints; we refer to the introduction for its brief intuitive outline.

▶ **Theorem 18.** *Let $T$ be a semi-complete digraph. Then any outdegree ordering of $V(T)$ has width at most $m(\mathbf{ctw}(T))$, where $m(t) = 100t^2 + 22t + 1$.*

**Proof.** Let $\sigma$ be any outdegree ordering of $V(T)$. If $\sigma$ had width more than $m(\mathbf{ctw}(T))$, then one of the partitions $(\sigma[\alpha], V(T) \setminus \sigma[\alpha])$ would be a $(m(\mathbf{ctw}(T)) + 1)$-backward tangle. Existence of such a structure is a contradiction with Lemma 16. ◀

This gives raise to a straightforward approximation algorithm for cutwidth of a semi-complete digraph that simply sorts the vertices with respect to outdegrees, and then scans through the ordering checking whether it has small width. Note that this scan may be performed in $O(|V(T)|^2)$ time, as we maintain the cut between the prefix and the suffix of the ordering by iteratively moving one vertex from the suffix to the prefix.

▶ **Theorem 19.** *There exists an algorithm which, given a semi-complete digraph $T$ and an integer $k$, in time $O(|V(T)|^2)$ outputs an ordering of $V(T)$ of width at most $m(k)$ or a $(m(k) + 1)$-backward tangle in $T$, where $m(t) = 100t^2 + 22t + 1$. In the second case the algorithm concludes that $\mathbf{ctw}(T) > k$.*

▶ **Theorem 20 (♠).** *There exists an algorithm, which given a semi-complete digraph $T$ and an integer $k$, in time $O(2^{O(k)}|V(T)|^2)$ outputs an ordering of $V(T)$ of width at most $k$, or correctly concludes that $\mathbf{ctw}(T) > k$.*

## 5 Algorithms for pathwidth

### 5.1 Subset selectors for bipartite graphs

In this section we propose a formalism for expressing selection of a subset of vertices of a bipartite graph. We introduce this formal layer as in the approximation and exact algorithms for pathwidth we use two different such concepts that share some properties. In this extended abstract we present only the one used in the approximation algorithm; the second one is deferred to the full version of the paper together with the description of the exact algorithm.

Let $\mathcal{B}$ be the class of undirected bipartite graphs with fixed bipartition, expressed as triples: left side, right side, the edge set. Let $\mu(G)$ be the size of a maximum matching in $G$.

▶ **Definition 21.** A function $f$ defined on $\mathcal{B}$ is called a *subset selector* if $f(G) \subseteq V(G)$ for every $G \in \mathcal{B}$. A *reversed* subset selector $f^{\mathrm{rev}}$ is defined as $f^{\mathrm{rev}}((X, Y, E)) = f((Y, X, E))$. We say that subset selector $f$ is

- a *vertex cover selector* if $f(G)$ is a vertex cover of $G$ for every $G \in \mathcal{B}$, i.e., every edge of $G$ has at least one endpoint in $f(G)$;
- *symmetric* if $f = f^{\mathrm{rev}}$;
- *monotonic* if for every graph $G = (X, Y, E)$ and its subgraph $G' = G \setminus w$ where $w \in Y$, we have that $f(G) \cap X \supseteq f(G') \cap X$ and $f(G) \cap (Y \setminus \{w\}) \subseteq f(G') \cap Y$.

The following observation expresses, how monotonic subset selectors behave with respect to modifications of the graph. By addition of a vertex we mean adding a new vertex to the vertex set, together with an arbitrary set of edges connecting it to the old ones.

▶ **Lemma 22 (♠).** *Assume that $f$ and $f^{\mathrm{rev}}$ are monotonic subset selector and let $G = (X, Y, E)$ be a bipartite graph.*

- *If $v \in f(G) \cap Y$ then $v$ stays chosen by $f$ after any sequence of additions of vertices to the left side and deletions of vertices (different from $v$) from the right side.*
- *If $v \in X \setminus f(G)$ then $v$ stays not chosen by $f$ after any sequence of additions of vertices to the left side and deletions of vertices from the right side.*

The subset selector that will be used for the approximation of pathwidth is the following:

▶ **Definition 23.** By *matching selector* $\mathfrak{M}$ we denote a subset selector that assigns to every bipartite graph $G$ the set of all the vertices of $G$ that are matched in **every** maximum matching in $G$.

Let us note that for any bipartite graph $G = (X, Y, E)$ we have that $|\mathfrak{M}(G) \cap X|, |\mathfrak{M}(G) \cap Y| \leq \mu(G)$. It appears that $\mathfrak{M}$ is a symmetric and monotonic vertex cover selector. The symmetry is obvious. The crucial property of $\mathfrak{M}$ is monotonicity: its proof requires technical and careful analysis of alternating and augmenting paths in bipartite graphs. $\mathfrak{M}$ admits also an alternative characterization (whose details may be found in the full version of the paper): it can be computed directly from any maximum matching by considering alternating paths originating in unmatched vertices. This observation can be utilized to construct an algorithm that maintains $\mathfrak{M}(G)$ efficiently during graph modifications. Moreover, from this alternative characterization it is clear that $\mathfrak{M}$ is a vertex cover selector. The following lemma expresses all the vital properties of $\mathfrak{M}$ that will be used in the approximation algorithm for pathwidth.

▶ **Lemma 24 (♠).** *$\mathfrak{M}$ is a symmetric, monotonic vertex cover selector, which can be maintained together with a maximum matching of $G$ with updates times $O((\mu(G)+1) \cdot |V(G)|)$ during vertex additions and deletions. Moreover, $|\mathfrak{M}(G) \cap X|, |\mathfrak{M}(G) \cap Y| \leq \mu(G)$ for every bipartite graph $G = (X, Y, E)$.*

## 5.2 Algorithms

In this section we present the algorithms for computing pathwidth. We aim at explaining the approximation algorithm in details, as it is the main result of this paper. Due to the space constraints, the description of the exact algorithm is deferred to the full version; we refer to the introduction for its brief intuitive outline. We introduce the approximation algorithm with an additional parameter $\ell$; taking $\ell = 5k$ gives the promised 7-approximation, but modifying the parameter $\ell$ may be useful to improve quality of the obtained degree tangle.

▶ **Theorem 25.** *There exists an algorithm, which given a semi-complete digraph $T$ and integers $k$ and $\ell \geq 5k$, in time $O(k|V(T)|^2)$ outputs one of the following:*

- *an $(\ell + 2, k)$-degree tangle in $T$;*
- *a $(k + 1, k)$-matching tangle in $T$;*
- *a path decomposition of $T$ of width at most $\ell + 2k$.*

*In the first two cases the algorithm can correctly conclude that $\mathbf{pw}(T) > k$.*

**Proof.** The last sentence follows from Lemmas 10 and 13. We proceed to the algorithm.

The algorithm first computes any outdegree ordering $\sigma = (v_1, v_2, \ldots, v_n)$ of $V(T)$ in $O(|V(T)|^2)$ time, where $n = |V(T)|$. Then in $O(|V(T)|)$ time we check if there is an index $i$

such that $d^+(v_{i+\ell+1}) \leq d^+(v_i) + k$. If this is true, then $\{v_i, v_{i+1}, \ldots, v_{i+\ell+1}\}$ is an $(\ell+2, k)$-degree tangle which can be safely output by the algorithm. From now on we assume that such a situation does not occur, i.e, $d^+(v_{i+\ell+1}) > d^+(v_i) + k$ for every index $i$.

We define a separation sequence $R_0 = ((A_0, B_0), (A_1, B_1), \ldots, (A_{n-\ell}, B_{n-\ell}))$ as follows. Let $S_i^0 = \{v_{i+1}, v_{i+2}, \ldots, v_{i+\ell}\}$ and $H_i = (X_i, Y_i, E_i)$ be a bipartite graph with $X_i = \{v_1, \ldots, v_i\}$, $Y_i = \{v_{i+\ell+1}, v_{i+\ell+2}, \ldots, v_n\}$ and $xy \in E_i$ if and only if $(x, y) \in E(T)$. If $\mu(H_i) > k$, then vertices matched in a maximum matching of $H_i$ form a $(k+1, k)$-matching tangle in $T$, which can be safely output by the algorithm. Otherwise, let $S_i = S_i^0 \cup \mathfrak{M}(H_i)$ and we set $A_i = X_i \cup S_i$ and $B_i = Y_i \cup S_i$; the fact that $(A_i, B_i)$ is a separation follows from the fact that $\mathfrak{M}$ is a vertex cover selector. Finally, we add separations $(\emptyset, V(T))$ and $(V(T), \emptyset)$ at the ends of the sequence, thus obtaining separation sequence $R$. We claim that $R$ is a separation chain. Note that if we prove it, the width of the corresponding path decomposition is upper bounded by $\max_{0 \leq i \leq n-\ell-1} |\{v_{i+1}, v_{i+2}, \ldots, v_{i+\ell+1}\} \cup (\mathfrak{M}(H_i) \cap X_i) \cup (\mathfrak{M}(H_{i+1}) \cap Y_{i+1})| - 1 \leq \ell + 1 + 2k - 1 = \ell + 2k$, by monotonicity of $\mathfrak{M}$.

It suffices to show that for every $i$ we have that $A_i \subseteq A_{i+1}$ and $B_i \supseteq B_{i+1}$. This, however, follows from Lemma 22 and the fact that $\mathfrak{M}$ is monotonic. $H_{i+1}$ differs from $H_i$ by deletion of one vertex on the right side and addition of one vertex on the left side, so we have that $A_{i+1}$ differs from $A_i$ only by possibly incorporating vertex $v_{i+\ell+1}$ and some vertices from $Y_{i+1}$ that became chosen by $\mathfrak{M}$, and $B_{i+1}$ differs from $B_i$ only by possibly losing vertex $v_{i+1}$ and some vertices from $X_i$ that ceased to be chosen by $\mathfrak{M}$.

Separation chain $R$ can be computed in $O(k|V(T)|^2)$ time: we consider consecutive sets $S_i^0$ and maintain the graph $H_i$ together with a maximum matching in it and $\mathfrak{M}(H_i)$. As going to the next set $S_i^0$ can be modelled by one vertex deletion and one vertex additions in graph $H_i$, by Lemma 24 we have that the time needed for an update is $O(k|V(T)|)$; note that whenever the size of the maximum matching exceeds $k$, we terminate the algorithm by outputting the obtained matching tangle. As we make $O(|V(T)|)$ updates, the time bound follows. Translating a separation chain into a path decomposition can be done in $O(\ell|V(T)|)$ time, assuming that we store the separators along with the separations. ◀

▶ **Theorem 26 (♠).** *There exists an algorithm that, given a semi-complete digraph $T$ and an integer $k$, in $O(2^{O(k \log k)}|V(T)|^2)$ time computes a path decomposition of $T$ of width at most $k$, or correctly concludes that no such exists.*

## 6 Topological containment and immersion

▶ **Theorem 27.** *There exists an algorithm which, given a semi-complete $T$ and a digraph $H$ with $k = |H|$, in time $O(2^{O(k \log k)}|V(T)|^2)$ checks whether $H$ is topologically contained in $T$.*

**Proof.** We run the algorithm given by Theorem 25 for parameters $20k$ and $520k$, which either returns a $(520k+2, 20k)$-degree tangle, a $(20k+1, 20k)$-matching tangle or a decomposition of width at most $560k$. If the last is true, we run the dynamic programming routine, which works in $O(2^{O(k \log k)}|V(T)|)$ time. The routine can be constructed in a similar manner to the one for immersion in [5], and its full description appears in the full version of the paper. However, if the approximation algorithm returned an obstacle, by Lemmas 11, 14 and 8 we can provide a positive answer: existence of a $(520k+2, 20k)$-degree tangle or a $(20k+1, 20k)$-matching tangle ensures that $H$ is topologically contained in $T$. ◀

By plugging in the dynamic programming routine for immersion of [5] instead of topological containment, we obtain the following.

▶ **Theorem 28.** *There exists an algorithm which given a semi-complete $T$ and a digraph $H$ with $k = |H|$, in time $O(2^{O(k^2 \log k)}|V(T)|^2)$ checks whether $H$ can be immersed into $T$.*

Finally, we can reduce the polynomial factor in the running time of the algorithm for rooted immersion of [5] by substituting the approximation routine for pathwidth with the one given by Theorem 25. However, one needs to be careful with running times of all the components of the algorithm; therefore, the proof is included in the full version of the paper.

▶ **Theorem 29 (♠).** *There exists an algorithm which given a rooted semi-complete $\mathbf{T}$ and a rooted digraph $\mathbf{H}$ with $k = |H|$, in time $O(f(k)|V(T)|^3)$ checks whether $\mathbf{H}$ is a rooted immersion of in $\mathbf{T}$, for some elementary function $f$.*

## 7 Conclusions

The natural question stemming from this work is whether the new set of obstacles can give raise to more powerful irrelevant vertex rules. For example, if we consider the Rooted Immersion problem, it is tempting to try to replace finding an irrelevant vertex in a triple, as presented in [5] (see this work for an exact definition of a triple), with a direct irrelevant vertex rule on a short jungle of size polynomial in the size of the digraph to be immersed. If this was possible, the running time for the algorithm for Rooted Immersion could be trimmed to single-exponential in terms of the size of the digraph to be immersed.

**Acknowledgements** The author thanks Marek Cygan, Fedor V. Fomin, and Marcin Pilipczuk for helpful discussions about this project.

## References

1 Jonathan F. Buss and Judy Goldsmith. Nondeterminism within P. *SIAM J. Comput.*, 22(3):560–572, 1993.
2 Maria Chudnovsky, Alexandra Fradkin, and Paul Seymour. Tournament immersion and cutwidth. *J. Comb. Theory Ser. B*, 102(1):93–101, 2012.
3 Maria Chudnovsky, Alex Scott, and Paul Seymour. Vertex disjoint paths in tournaments, 2011. Manuscript.
4 Maria Chudnovsky and Paul D. Seymour. A well-quasi-order for tournaments. *J. Comb. Theory, Ser. B*, 101(1):47–53, 2011.
5 Fedor V. Fomin and Michał Pilipczuk. Jungles, bundles, and fixed parameter tractability. *CoRR*, abs/1112.1538, 2011. To appear in proceedings of SODA 2013.
6 Steven Fortune, John E. Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.
7 Alexandra Fradkin and Paul Seymour. Edge-disjoint paths in digraphs with bounded independence number, 2010. Manuscript.
8 Alexandra Fradkin and Paul Seymour. Tournament pathwidth and topological containment, 2011. Manuscript.
9 Robert Ganian, Petr Hliněný, Joachim Kneis, Daniel Meister, Jan Obdržálek, Peter Rossmanith, and Somnath Sikdar. Are there any good digraph width measures? In *IPEC 2010*, volume 6478 of *Lecture Notes in Computer Science*, pages 135–146. Springer, 2010.
10 Ilhee Kim and Paul Seymour. Tournament minors. *CoRR*, abs/1206.3135, 2012.
11 Michał Pilipczuk. Computing cutwidth and pathwidth of semi-complete digraphs via degree orderings. *CoRR*, abs/1210.5363, 2012.
12 Dimitrios M. Thilikos, Maria J. Serna, and Hans L. Bodlaender. Cutwidth I: A linear time fixed parameter algorithm. *J. Algorithms*, 56(1):1–24, 2005.