Degreewidth: A New Parameter for Solving Problems on Tournaments

Tom Davot, Lucas Isenmann, Sanjukta Roy, Jocelyn Thiebaut

Degreewidth: a New Parameter for Solving Problems on Tournaments*

Tom Davot[1], Lucas Isenmann[2], Sanjukta Roy[3], and Jocelyn Thiebaut[4]

[1] Université de Technologie de Compiègne, CNRS, Heudiasyc, France
tom.davot@hds.utc.fr
[2] Université de Montpellier, France lucas.isenmann@umontpellier.fr
[3] Pennsylvania State University sanjukta@psu.edu
[4] Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic jocelyn.thiebaut@cvut.cz

Abstract. In the paper, we define a new parameter for tournaments called degreewidth which can be seen as a measure of how far is the tournament from being acyclic. The degreewidth of a tournament T denoted by $\Delta(T)$ is the minimum value k for which we can find an ordering $\langle v_1, \ldots, v_n \rangle$ of the vertices of T such that every vertex is incident to at most k backward arcs (*i.e.* an arc (v_i, v_j) such that $j < i$). Thus, a tournament is acyclic if and only if its degreewidth is zero. Additionally, the class of sparse tournaments defined by Bessy *et al.* [ESA 2017] is exactly the class of tournaments with degreewidth one.
We study computational complexity of finding degreewidth. We show it is NP-hard and complement this result with a 3-approximation algorithm. We provide a $O(n^3)$-time algorithm to decide if a tournament is sparse, where n is its number of vertices.
Finally, we study classical graph problems Dominating Set and Feedback Vertex Set parameterized by degreewidth. We show the former is fixed-parameter tractable whereas the latter is NP-hard even on sparse tournaments. Additionally, we show polynomial time algorithm for Feedback Arc Set on sparse tournaments.

Keywords: Tournaments · NP-hardness · graph-parameter · feedback arc set · approximation algorithm · parameterized algorithms

1 Introduction

A tournament is a directed graph such that there is exactly one arc between each pair of vertices. Tournaments form a very rich subclass of digraphs which has been widely studied both from structural and algorithmic point of view [4].

* Preprint version. The editor version can be found at https://doi.org/10.1007/978-3-031-43380-1_18.
Sanjukta Roy was affiliated to Faculty of Information Technology, Czech Technical University in Prague when majority of this work was done. Jocelyn Thiebaut was supported by the CTU Global postdoc fellowship program.

Unlike for complete graphs, a number of classical problems remain difficult in tournaments and therefore interesting to study. These problems include Dominating Set [14], Winner Determination [22], or maximum cycle packing problems. For example, Dominating Set is W[2]-hard on tournaments with respect to solution size [14]. However, many of these problems become easy on acyclic tournaments (*i.e.* without directed cycle). Therefore, a natural question that arises is whether these problems are easy to solve on tournaments that are close to being acyclic. The phenomenon of a tournament being "close to acyclic" can be captured by minimum size of a *feedback arc set* (fas). A fas is a collection of arcs that, when removed from the digraph (or, equivalently, reversed) makes it acyclic. This parameter has been widely studied, for numerous applications in many fields, such as circuit design [19], or artificial intelligence [5, 13]. However, the problem of finding a minimum fas on tournaments (the problem is then called *FAST* for Feedback Arc Set in Tournaments), remained opened for over a decade before being proven NP-complete [3, 10]. From the approximability point of view, van Zuylen and Williamson [25] provided a 2-approximation of FAST, and Kenyon-Mathieu and Schudy [21] a PTAS algorithm. On the parameterized-complexity side, Feige [15] as well as Karpinski and Schudy [20] independently proved an $2^{O(\sqrt{k})} + n^{O(1)}$ running-time algorithm. Another way to define FAST is to consider the problem of finding an ordering of the vertices $\langle v_1, \ldots, v_n \rangle$ minimising the number of arcs (v_i, v_j) with $j < i$; such arcs are called *backward arcs*. Then, it is easy to see that a tournament is acyclic if and only if it admits an ordering with no backward arcs. Several parameters exploiting an ordering with specific properties have been studied in this sense [18] such as the cutwidth. Given an ordering of vertices, for each prefix of the ordering we associate a cut defined as the set of backward arcs with head in the prefix and tail outside of it. Then cutwidth is the minimum value, among all the orderings, of the maximum size of any possible cut w.r.t the ordering (a formal definition is introduced in next section). It is well-known that computing cutwidth is NP-complete [17], and has an $O(\log^2(n))$-approximation on general graphs [23]. Specifically on tournaments, one can compute an optimal ordering for the cutwidth by sorting the degrees according to the in-degrees [16].

In this paper, we propose a new parameter called *degreewidth* using the concept of backward arcs in an ordering of vertices. Degreewidth of a tournament is the minimum value, among all the orderings, of the maximum number of backward arcs incident to a vertex. Hence, an acyclic tournament is a tournament with degreewidth zero. Furthermore, one can notice that tournaments with degreewidth at most one are the same as the *sparse tournaments* introduced in [8, 24]. A tournament is *sparse* if there exists an ordering of vertices such that the backward arcs form a matching. It is known that computing a maximum sized arc-disjoint packing of triangles and computing a maximum sized arc-disjoint packing of cycles can be done in polynomial time [7] on sparse tournaments.

To the best of our knowledge this paper is the first to study the parameter degreewidth. As we will see in the next part, although having similarities with the cutwidth, this new parameter differs in certain aspects. We first study structural and computational aspects of degreewidth. Then, we show how it can be used to solve efficiently some classical problems on tournaments.

Our contributions and organization of the paper Next section provides the formal definition of degreewidth and some preliminary observations. In Section 3, we first study the degreewidth of a special class of tournaments, called regular tournaments, of order $2k+1$ and prove they have degreewidth k. We then prove that it is NP-hard to compute the degreewidth in general tournaments. We finally give a 3-approximation algorithm to compute this parameter which is tight in the sense that it cannot produce better than 3-approximation for a class of tournaments.

Then in Section 4, we focus on tournaments with degreewidth one, *i.e.*, the sparse tournaments. Note that it is claimed in [8] that there exists a polynomial-time algorithm for finding such ordering, but the only available algorithm appearing in [24, Lemma 35.1, p.97] seems to be incomplete (see discussion Subsection 4.2). We first define a special class of tournaments that we call U-tournaments. We prove there are only two possible sparse orderings for such tournaments. Then, we give a polynomial time algorithm to decide if a tournament is sparse by carefully decomposing it into U-tournaments.

Finally, in Section 5 we study degreewidth as a parameter for some classical graph problems. First, we show an FPT algorithm for Dominating Set w.r.t degreewidth. Then, we focus on tournaments with degreewidth one. We design an algorithm running in time $O(n^3)$ to compute a Feedback Arc Set on tournaments on n vertices with degreewidth one. However, we show that Feedback Vertex Set remains NP-complete on this class of tournaments.

Due to paucity of space the missing proofs are deferred to full version [12].

2 Preliminaries

2.1 Notations

In the following, all the digraphs are simple, that is without self-loop and multiple arcs sharing the same head and tail, and all cycles are directed cycles. The *underlying graph* of a digraph D is an undirected graph obtained by replacing every arc of D by an edge. Furthermore, we use $[n]$ to denote the set $\{1, 2, \dots, n\}$.

A tournament is a digraph where there is exactly one arc between each pair of vertices. It can alternatively be seen as an orientation of the complete graph. Let T be a tournament with vertex set $\{v_1, \dots, v_n\}$. We denote $N^+(v)$ the *out-neighbourhood* of a vertex v, that is the set $\{u \mid (v, u) \in A(T)\}$. Then, T being a tournament, the *in-neighbourhood* of the vertex v denoted $N^-(v)$ corresponds to $V(T) \setminus (N^+(v) \cup \{v\})$. The *out-degree* (resp. *in-degree*) of v denoted $d^+(v)$ (resp. $d^-(v)$) is the size of its out-neighbourhood (resp. in-neighbourhood).

A tournament T of order $2k+1$ is *regular* if for any vertex v, we have $d^+(v) = d^-(v) = k$. Let X be a subset of $V(T)$. We denote by $T - X$ the subtournament induced by the vertices $V(T) \setminus X$. Furthermore, when X contains only one vertex $\{v\}$ we simply write $T - v$ instead of $T - \{v\}$. We also denote by $T[X]$ the tournament induced by the vertices of X. Finally, we say that $T[X]$

dominates T if, for every $x \in X$ and every $y \in V(T) \setminus X$, we have $(x, y) \in A(T)$. For more definitions on directed graphs, please refer to [4].

Given a tournament T, we equip the vertices of T with is a strict total order $\prec_\sigma$. This operation also defines an ordering of the set of vertices denoted by $\sigma := \langle v_1, \ldots, v_n \rangle$ such that $v_i \prec_\sigma v_j$ if and only if $i < j$. Given two distinct vertices u and v, if $u \prec_\sigma v$ we say that u is *before* v in σ; otherwise, u is *after* v in σ. Additionally, an arc (u, v) is said to be *forward* (resp. *backward*) if $u \prec_\sigma v$ (resp. $v \prec_\sigma u$). A topological ordering is an ordering without any backward arcs. A tournament that admits a topological ordering does not contain a cycle. Hence, it is said to be *acyclic*.

A *pattern* $p_1 := \langle v_1, \ldots, v_k \rangle$ is a sequence of vertices that are consecutive in an ordering. Furthermore, considering a second pattern $p_2 := \langle u_1, \ldots, u_{k'} \rangle$ where $\{v_1, \ldots, v_k\}$ and $\{u_1, \ldots, u_{k'}\}$ are disjoint, the pattern $\langle p_1, p_2 \rangle$ is defined by $\langle v_1, \ldots, v_k, u_1, \ldots, u_{k'} \rangle$.

Degreewidth Given a tournament T, an ordering σ of its vertices $V(T)$ and a vertex $v \in V(T)$, we denote $d_\sigma(v)$ to be the number of backward arcs incident to v in σ, that is $d_\sigma(v) := |\{u \mid u \prec_\sigma v,\, u \in N^+(v)\} \cup \{u \mid v \prec_\sigma u,\, u \in N^-(v)\}|$. Then, we define the degreewidth of a tournament with respect to the ordering σ, denoted by $\Delta_\sigma(T) := \max\{d_\sigma(v) \mid v \in V(T)\}$. Note that $\Delta_\sigma(T)$ is also the maximum degree of the underlying graph induced by the backward arcs of σ. Finally, we define the degreewidth $\Delta(T)$ of the tournament T as follows.

Definition 1. *The degreewidth of a tournament T, denoted $\Delta(T)$, is defined as $\Delta(T) := \min_{\sigma \in \Sigma(T)} \Delta_\sigma(T)$, where $\Sigma(T)$ is the set of possible orderings for $V(T)$.*

As mentioned before, this new parameter tries to measure how far a tournament is from being acyclic. Indeed, it is easy to see that a tournament T is acyclic if and only if $\Delta(T) = 0$. Additionally, when degreewidth of a tournament is one, it coincides with the notion of sparse tournaments, introduced in [8].

Remark. The definition of degreewidth naturally extends to directed graphs and we hope it will be an exciting parameter for problems on directed graphs. However, in this article we study this as a parameter for tournaments which is well-studied in various domains [2, 9, 22]. Moreover, degreewidth also gives a succinct representation of a tournament. Informally, sparse graphs[5] are graphs with a low density of edges. Hence, it may be surprising to talk about sparsity in tournaments. However, if a tournament on n vertices admits an ordering σ where the backward arcs form a matching, then it can be encoded by σ and the set of backward arcs (at most $n/2$). Thus, the size of the encoding for such tournament is $O(n)$, instead of $O(n^2)$. For a tournament with degreewidth k, the same reasoning implies that it can be encoded in $O(kn)$ space.

2.2 Links to other parameters

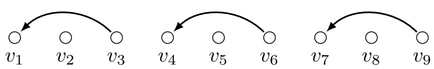
Feedback arc/vertex set A *feedback arc set* (fas) is a collection of arcs that, when removed from the digraph (or, equivalently, reversed) makes it acyclic. The

[5] Not to be confused with sparse tournaments that has an arc between every pair of vertices, hence, is not a sparse graph.

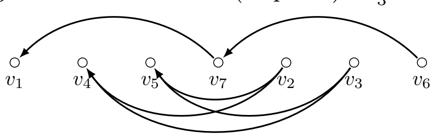
size of a minimum fas is considered for measuring how far the digraph is from being acyclic. In this context, degreewidth comes as a promising alternative. Finding a small subset of arcs hitting all substructures (in this case, directed cycles) of a digraph is one of the fundamental problems in graph theory. Note that we can easily bound the degreewidth of a tournament by its minimum fas f.

Observation 1. *For any tournament T, we have $\Delta(T) \leq |f|$.*

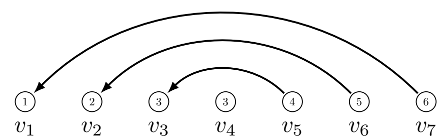
Note however that the opposite is not true; it is possible to construct tournaments with small degreewidth but large fas, see Figure 1(a).



(a) Example of a tournament with degreewidth one but fas (resp. fvs) $\frac{|V(T)|}{3}$.



(b) Example of a tournament T with fvs one (v_7) but degreewidth $\frac{|V(T)|-3}{2}$. The topological ordering of $T - v_7$ is $\langle v_1, v_2, v_3, v_4, v_5, v_6 \rangle$.



(c) Example of a tournament with degreewidth one but cutwidth $\frac{|V(T)|-1}{2}$. Since the vertices are sorted by increasing in-degrees (values inside the vertices), this is an optimal ordering for the cutwidth.

Fig. 1 Link between degreewidth and other parameters. All the non-depicted arcs are forward.

Similarly, a *feedback vertex set* (fvs) consists of a collection of vertices that, when removed from the digraph makes it acyclic. However, – unlike the feedback arc set – the link between feedback vertex set and degreewidth seems less clear; we can easily construct tournaments with low degreewidth and large fvs (see Figure 1(a)) as well as large degreewidth and small fvs (see Figure 1(b)).

Cutwidth Let us first recall the definition of the cutwidth of a digraph. Given an ordering $\sigma := \langle v_1, \ldots, v_n \rangle$ of the vertices of a digraph D, we say that a prefix of σ is a sequence of consecutive vertices $\langle v_1, \ldots, v_k \rangle$ for some $k \in [n]$. We associate for each prefix of σ a *cut* defined as the set of backward arcs with head in the prefix and tail outside of it. The *width* of the ordering σ is defined as the size of a maximum cut among all the possible prefixes of σ. The cutwidth of D, $ctw(D)$, is the minimum width among all orderings of the vertex set of D.

Intuitively, the difference between the cutwidth and the degreewidth is that the former focuses on the backward arcs going "above" the intervals between the vertices while the latter focuses on the backward arcs coming from and to the vertices themselves. Observe that for any tournament T, the degreewidth is bounded by a function of the cutwidth. Formally, we have the following

Observation 2. *For any tournament T, we have $\Delta(T) \leq 2ctw(T)$.*

Note however that the opposite is not true; it is possible to construct tournaments with small degreewidth but large cutwidth, see Figure 1(c). We remark that the graph problems that we study parameterized by degreewidth, namely, minimising fas, fvs, and dominating set are FPT w.r.t cutwidth [1, 11].

3 Degreewidth

In this section, we present some structural and algorithmical results for the computation of degreewidth. We first introduce the following lemma that provides a lower bound on the degreewith.

Lemma 1. *Let T be a tournament. Then $\Delta(T) \geq \min_{v \in V(T)} d^-(v)$ and $\Delta(T) \geq \min_{v \in V(T)} d^+(v)$.*

3.1 Degreewidth of regular tournaments

Theorem 1. *Let T be a regular tournament of order $2k+1$. Then $\Delta(T) = k$. Furthermore, for any ordering σ, by denoting u and v respectively the first and last vertices in σ, we have $d_\sigma(u) = d_\sigma(v) = k$.*

Note that regular tournaments contain many cycles; therefore it is not surprising that their degreewidth is large. This corroborates the idea that this parameter measures how far a tournament is from being acyclic.

3.2 Computational complexity

We now show that computing the degreewidth of a tournament is NP-hard by defining a reduction from Balanced 3-SAT(4), proven NP-complete [6] where each clause contains exactly three unique literals and each variable occurs two times positively and two times negatively.

Let φ be a Balanced 3-SAT(4) formula with m clauses $c_1, \ldots, c_m$ and n variables $x_1, \ldots, x_n$. In the construction, we introduce several regular tournaments of size W or $\frac{W+1}{2} + n + m$, where W is value greater than $n^3 + m^3$. Note that $n + m$ is necessarily odd since $4n = 3m$. By taking a value $W = 3 \mod 4$, we ensure that every regular tournament of size W or $\frac{W+1}{2} + n + m$ has an odd number of vertices.

Construction 1. *Let φ be a* Balanced 3-SAT(4) *formula with m clauses $c_1, \ldots, c_m$ clauses and n variables $x_1, \ldots, x_n$ such that n is odd and m is even. Let $W = 3 \mod 4$ be an integer greater than $n^3 + m^3$. We construct a tournament T as follows.*

- *Create two regular tournaments A and D of order $\frac{W+1}{2} + m + n$ such that D dominates A.*
- *Create two regular tournaments B and C of order W such that A dominates $B \cup C$, B dominates C and $B \cup C$ dominates D.*
- *Create an acyclic tournament X of order $2n$ with topological ordering $\langle v_1, v_1', \ldots, v_n, v_n' \rangle$ such that $A \cup C$ dominates X and X dominates $B \cup D$.*

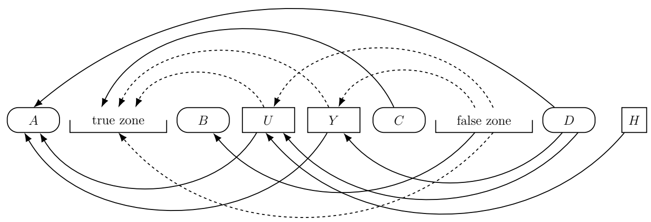


Fig. 2 Example of a nice ordering. A rectangle represents an acyclic tournament, while a rectangle with rounded corners represents a regular tournament. A plain arc between two patterns P and P' represents the fact that there is a backward arc between every pair of vertices $v \in P$ and $v' \in P'$. A dashed arc means some backward arcs may exist between the patterns.

- *Create an acyclic tournament Y of order $2m$ with topological ordering $\langle q_1, q'_1, \dots, q_m, q'_m \rangle$ such that $B \cup D$ dominates Y and Y dominates $A \cup C$.*
- *For each clause c_ℓ and each variable x_i of φ,*
  - *if x_i occurs positively in c_ℓ, then $\{v_i, v'_i\}$ dominates $\{q_\ell, q'_\ell\}$,*
  - *if x_i occurs negatively in c_ℓ, then $\{q_\ell, q'_\ell\}$ dominates $\{v_i, v'_i\}$,*
  - *if x_i does not occur in c_ℓ, then introduce the paths (v_i, q_ℓ, v'_i) and (v'_i, q'_ℓ, v_i).*
- *Introduce an acyclic tournament $U = \{u_i^p, \bar{u}_i^p \mid i \leq n, p \leq 2\}$ of order $4n$ such that U dominates $A \cup Y \cup C$ and $B \cup D$ dominates U. For each variable x_i, add the following paths,*
  - *for all variable $x_k \neq x_i$ and all $p \leq 2$, introduce the paths (v_k, u_i^p, v'_k) and $(v'_k, \bar{u}_i^p, v_k)$,*
  - *introduce the paths (v_i, u_i^1, v'_i), (v'_i, u_i^2, v_i), $(v_i, \bar{u}_i^1, v'_i)$ and $(v'_i, \bar{u}_i^2, v_i)$.*
- *Finally, introduce an acyclic tournament $H = \{h_1, h_2\}$ with topological ordering $\langle h_1, h_2 \rangle$ and such that $A \cup B \cup C \cup X \cup Y \cup D$ dominates H and H dominates U.*

We call a vertex of X a *variable vertex* and a vertex of Y a *clause vertex*. Furthermore, we say that the vertices (v_i, v'_i) (resp. (q_ℓ, q'_ℓ)) is a *pair of variable vertices* (resp. *pair of clause vertices*).

Definition 2. *Let T be a tournament resulting from Construction 1. An ordering σ of T is nice if:*
- $\Delta_\sigma(A) = \frac{|A|-1}{2}$, $\Delta_\sigma(B) = \frac{|B|-1}{2}$, $\Delta_\sigma(C) = \frac{|C|-1}{2}$, *and* $\Delta_\sigma(D) = \frac{|D|-1}{2}$,
- *σ respects the topological ordering of $U \cup Y$,*
- $A \prec_\sigma B \prec_\sigma U \prec_\sigma Y \prec_\sigma C \prec_\sigma D \prec_\sigma H$, *and*
- *for any variable x_i, either $A \prec_\sigma v_i \prec_\sigma v'_i \prec_\sigma B$ or $C \prec_\sigma v_i \prec_\sigma v'_i \prec_\sigma D$.*

An example of a nice ordering is depicted in Figure 2. Let σ be a nice ordering, we call the pattern corresponding to the vertices between A and B, the *true zone* and the pattern after the vertices of C the *false zone*. Let (q_ℓ, q'_ℓ) be a pair of clause vertices and let (v_i, v'_i) be a pair of variable vertices such that x_i occurs positively (resp. negatively) in c_ℓ in φ. We say that the pair (v_i, v'_i)

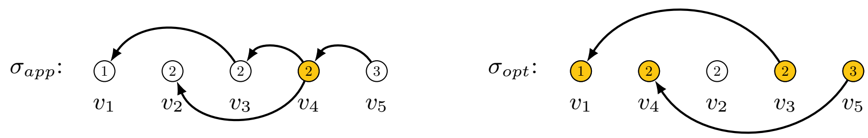


Fig. 3 Example of a tournament where the approximate algorithm can return an ordering σ_{app} (on the left) with degreewidth three while the optimal solution is one in σ_{opt} (on the right). Coloured vertices are the ones incident to the maximum number of backward arcs. all non-depicted arcs are forward arcs.

satisfies (q_ℓ, q'_ℓ) if v_i and v'_i both belong to the true zone (resp. false zone). Note that there is no backward arc between $\{q_\ell, q'_\ell\}$ and $\{v_i, v'_i\}$ if and only if (v_i, v'_i) satisfies (q_ℓ, q'_ℓ). Notice also that for any pair of variable vertices (v_i, v'_i) such that x_i does not appear in c_ℓ and (v_i, v'_i) is either in the true zone or in the false zone, then there is exactly two backward arcs between $\{q_\ell, q'_\ell\}$ and $\{v_j, v'_j\}$. Let φ be an instance of Balanced 3-SAT(4) and T its tournament resulting from Construction 1. We show that φ is satisfiable if and only if there exists an ordering σ of T such that $\Delta_\sigma(T) < W + 2m + 3n + 4$, which yields the following.

Theorem 2. *Given a tournament T and an integer k, it is NP-complete to compute an ordering σ of T such that $\Delta_\sigma(T) \leq k$.*

3.3 An approximation algorithm to compute degreewidth

In this subsection, we prove that sorting the vertices by increasing in-degree is a tight 3-approximation algorithm to compute the degreewidth of a tournament. Intuitively, the reasons why it returns a solution not too far from the optimal are twofold. Firstly, observe that the only optimal ordering for acyclic tournaments (*i.e.* with degreewidth 0) is an ordering with increasing in-degrees. Secondly, this strategy also gives an optimal solution for cutwidth in tournaments.

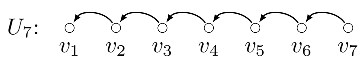
Theorem 3. *Ordering the vertices by increasing order of in-degree is a tight 3-approximation algorithm to compute the degreewidth of a tournament (see Figure 3).*

4 Results on sparse tournaments

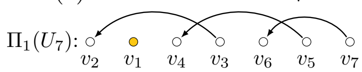
In this section, we focus on tournaments with degreewidth one, called sparse tournaments. The main result of this section is that unlike in the general case, it is possible to compute in polynomial time a sparse ordering of a tournament (if it exists). We begin with an observation about sparse orderings (if it exists).

Lemma 2. *Let T be a sparse tournament of order $n > 4$ and σ be an ordering of its vertices. If σ is a sparse ordering, then for any vertex v such that $d^-(v) = i$, the only possible positions of v in σ are $\{i, i+1, i+2\} \cap [n]$.*

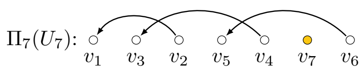
Note that Lemma 2 gives immediately an exponential running-time algorithm to decide if a tournament is sparse. However, we give in Subsection 4.2 a polynomial running-time algorithm for this problem. Before that we study a useful subclass of sparse tournaments, we call the U-tournaments.



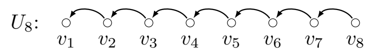
(a) The tournament U_7.



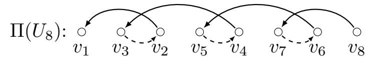
(b) The sparse ordering $\Pi_1(U_7)$. Note that v_1 is the only vertex not incident to any backward arc.



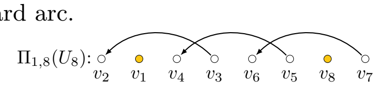
(c) The sparse ordering $\Pi_7(U_7)$. Note that v_7 is the only vertex not incident to any backward arc.



(d) The tournament U_8.



(e) The sparse ordering $\Pi(U_8)$. The dashed forward arcs is a minimum feedback arc set of the tournament. Note that all the vertices are incident to one backward arc.



(f) The sparse ordering $\Pi_{1,8}(U_8)$. Note that v_1 and v_8 are the only vertices not incident to any backward arc.

Fig. 4 The tournaments U_7 and U_8 and their sparse orderings. The non-depicted arcs are forward arcs.

4.1 *U*-tournaments

In this subsection, we study one specific type of tournaments called U-tournaments. Informally, they correspond to the acyclic tournaments where we reversed all the arcs of its Hamiltonian path.

Definition 3. *For any integer $n \geq 1$, we define U_n as the tournament on n vertices with $V(U_n) = \{v_1, v_2, \ldots, v_n\}$ and $A(U_n) = \{(v_{i+1}, v_i) \mid \forall i \in [n-1]\} \cup \{(v_i, v_j) \mid 1 \leq i < n, i+1 < j \leq n\}$. We say that a tournament of order n is a U-tournament if it is isomorphic to U_n.*

Figures 4(a) and 4(d) depict respectively the tournaments U_7 and U_8. This family of tournaments seems somehow strongly related to sparse tournaments and the following results will be useful later for both the polynomial algorithm to decide if a tournament is sparse and the polynomial algorithm for minimum feedback arc set in sparse tournaments. To do so, we prove that each U-tournament of order $n > 4$ has exactly two sparse orderings of its vertices that we formally define.

Definition 4. *Let $P(k) = \langle v_{k+1}, v_k \rangle$ be a pattern of two vertices of U_n for some integer $k \in [n-1]$. For any integer $n \geq 2$, we define the following special orderings of U_n:*
- *if n is even:*
  - *$\Pi(U_n)$ is the ordering given by $\langle v_1, P(2), P(4), \ldots, P(n-2), v_n \rangle$.*
  - *$\Pi_{1,n}(U_n)$ is the ordering given by $\langle P(1), P(3), \ldots, P(n-2), P(n) \rangle$.*
- *if n is odd:*
  - *$\Pi_1(U_n)$ is the ordering given by $\langle P(1), P(3), \ldots, P(n-2), v_n \rangle$.*
  - *$\Pi_n(U_n)$ is the ordering given by $\langle v_1, P(2), P(4), \ldots, P(n-3), P(n-1) \rangle$.*

Figures 4(b) and 4(c) (and Figures 4(e) and 4(f)) depict the orderings $\Pi_1(U_7)$ and $\Pi_7(U_7)$ (resp. $\Pi(U_8)$ and $\Pi_{1,8}(U_8)$) of the tournament U_7 (resp. U_8). One can notice that these orderings are sparse and the subscript of Π indicates the vertex (or vertices) without a backward arc incident to it in this ordering. In the following, we prove that when $n > 4$ there are no other sparse orderings of U_n. However, note that there are three possible sparse orderings of U_3 (namely, $\Pi_1(U_3)$ and $\Pi_3(U_3)$ defined previously, as well as $\Pi_2(U_3) := \langle v_3, v_2, v_1\rangle$) and three sparse orderings of U_4 (namely, $\Pi(U_4)$, $\Pi_{1,4}(U_4)$ as defined before, and $\Pi'(U_4) := \langle v_2, v_4, v_1, v_3\rangle$).

Theorem 4. *For each integer $n > 4$ there are exactly two sparse orderings of U_n. Specifically, if n is even, these two sparse orderings are $\Pi(U_n)$ and $\Pi_{1,n}(U_n)$; otherwise, the two sparse orderings are $\Pi_1(U_n)$ and $\Pi_n(U_n)$.*

4.2 A polynomial time algorithm for sparse tournaments

We give here a polynomial algorithm to compute a sparse ordering of a tournament (if any). First of all, let us recall a classical algorithm to compute a topological ordering of a tournament (if any): we look for the vertex v with the smallest in-degree; if v has in-degree one or more, we have a certificate that the tournament is not acyclic. Otherwise, we add v at the beginning of the ordering, and we repeat the reasoning on $T - v$, until $V(T)$ is empty.

The idea of the original "proof" in [24, Lemma 35.1, p.97] was similar: considering the set of vertices X of smallest in-degrees, put X at the beginning of the ordering, and remove X from the tournament. However, potential backward arcs from the remaining vertices of $V \setminus X$ to X may have been forgotten. For example, consider a tournament over 9 vertices consisting of a U_5 (with vertex set $\{v_1, \ldots, v_5\}$) that dominates a U_4 (with vertex set $\{u_1, ..., u_4\}$) except for the backward arc (u_4, v_5). It is sparse ($\langle \Pi_5(U_5), \Pi_{1,4}(U_4)\rangle$) but the algorithm returns the (non-sparse) ordering $\langle \Pi_1(U_5), \Pi_{1,4}(U_4)\rangle$ (v_5 is incident to two backward arcs). The problem is that this algorithm is too "local"; it will always prefer the sparse ordering $\Pi_1(U_{2k+1})$ over $\Pi_{2k+1}(U_{2k+1})$, but it may be necessary to take the latter. Therefore, to correct this, we needed a much more involved algorithm, requiring the study of the U-tournaments and the notion of quasi-domination (see Definition 6). Indeed, unlike the algorithm for the topological ordering, we may have to look more carefully how the vertices with low in-degrees are connected to the rest of the digraph. These correspond to the case where there exists a U-sub-tournament of T which either dominates or "quasi-dominates" (see Definition 6) the tournament T. Because of the latter possibility (where a backward arc (a, b) is forced to appear), we need to look for specific sparse orderings, called M-sparse orderings (where a or b should not be end-vertices of other backward arcs). As all the sparse orderings for U-tournaments have been described, we can derive a recursive algorithm.

Definition 5. *Let T be a tournament, X be a subset of vertices of T, and M be a subset of X. We say $T[X]$ is M-sparse if there exists an ordering σ of X such that $\Delta_{\sigma(T[X])}(X) \leq 1$ and $d_\sigma(v) = 0$ for all $v \in M$. In that case, σ is said to be an M-sparse ordering of $T[X]$.*

For example, $U_4[\{v_1, v_2, v_3\}]$ is $\{v_2\}$-sparse, because there exists a sparse ordering $\sigma := \langle v_3, v_2, v_1 \rangle$ of $U_4[\{v_1, v_2, v_3\}]$ such that $d_\sigma(v_2) = 0$. We remark that T is sparse if and only if T is $\emptyset$-sparse. In fact, the algorithm described in this section computes a $\emptyset$-sparse ordering of the given tournament (if any).

Definition 6 (see Figure 5). *Given a tournament T and two of its vertices a and b, we say that a subset of vertices X quasi-dominates T if:*
- *there exists an arc $(b, a) \in A(T)$ such that $a \in X$ and $b \notin X$,*
- *$(u, v) \in A(T)$ for every $(u, v) \in (X \times (V(T) \setminus X)) \setminus \{(a, b)\}$,*
- *$d^-(b) \geq |X| + 1$, and*
- *the vertex a has an out-neighbour in X.*

In this case, we also say X (b, a)-quasi-dominates T.

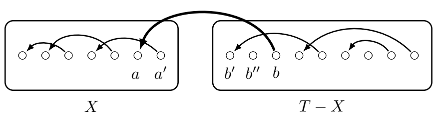


Fig. 5 An example where X (b, a)-quasi-dominates T. Non-depicted arcs are forward. The vertex a' is an out-neighbour of a in X, and b', b'' are in-neighbours of b in $T - X$.

We can create the algorithm `isUkMsparse` which given $(v_1, \dots, v_k)$ a U-tournament and M a subset of these vertices, returns a boolean which is True if and only if this tournament is M-sparse. We can also create the algorithm `getUsubtournament` which given T a tournament, and $X = (u_1, \dots, u_k)$ a list of vertices such that $d^-(u_1) = 1$ and $d^-(u_i) = i - 1$ and $(u_i, u_{i-1}) \in A(T)$ for all $i \in \{2, \dots, k\}$, returns a U-subtournament dominating or quasi-dominating T. With these two previous algorithms, we can derive Algorithm 3 `isMsparse`.

Algorithm 1: `getUsubtournament`

Data: T a tournament, and $X = (u_1, \dots, u_k)$ a list of vertices such that $d^-(u_1) = 1$ and $d^-(u_i) = i - 1$ and $(u_i, u_{i-1}) \in A(T)$ for all $i \in \{2, \dots, k\}$.
Result: A U-subtournament dominating or quasi-dominating T.

$w \longleftarrow$ a vertex of $N^-(u_k) \setminus X$;
if $d^-(w) = d^-(u_k)$ **then return** $X \cup \{w\}$ `/* this set dominates` T `*/` ;
else if $d^-(w) = d^-(u_k) + 1$ **then return** `getUsubtournament(`$T, X \cup \{w\}$`)` ;
else return X `/* this set` (w, u_k)`-quasi-dominates` T `*/` ;

Algorithm 2: `isUkMsparse`

Data: $(v_1, \dots, v_k)$ a U_k tournament, M a subset of the vertices of U_k
Result: `True` if U_k is M-sparse and `False` otherwise

if $k \leq 2$ **then return** *`True`* ;
else if $k = 3$ **then return** $|M| \leq 1$;
else if k *is even* **then return** $|M \setminus \{v_1, v_k\}| = 0$;
else if k *is odd* **then return** *($v_1 \notin M$ or $v_k \notin M$) and* $|M \setminus \{v_1, v_k\}| = 0$;

Theorem 5. *Algorithm 3 is correct. Hence, it is possible to decide if a tournament T with n vertices is sparse in $\mathcal{O}(n^3)$ by calling `isMsparse(`T,$\emptyset$`)`.*

Observe that we can easily modify Algorithm 3 to obtain a sparse ordering (if exists). Next corollary follows from the above algorithm.

Algorithm 3: `isMsparse`

Data: T a tournament, M a subset of the vertices of T
Result: `True` if T is M-sparse and `False` otherwise
if $|V(T)| \leq 1$ **then return** *True* ;
else if $\min_{v \in V(T)} d^-(v) \geq 2$ **then return** *False* ;
else if $\min_{v \in V(T)} d^-(v) = 0$ **then**
  $v \longleftarrow$ the vertex of in-degree 0;
  return `isMsparse`$(T - v, M \setminus \{v\})$;
else if $|\{v \in V(T) : d^-(v) = 1\}| = 1$ **then**
  $v, w \longleftarrow$ two vertices such that $d^-(v) = 1$ and $(w, v) \in A(T)$;
  return $v \notin M$ *and* `isMsparse`$(T - v, (M \cup \{w\}) \setminus \{v\})$;
else
  $v, w \longleftarrow$ two vertices of in-degree 1 such that $(w, v) \in A(T)$;
  $X \longleftarrow$ `getUsubtournament`$(T,(v, w))$;
  if X *dominates* T **then**
    return (`isUkMsparse`$(X, M \cap X)$ *and* `isMsparse`$(T - X, M \setminus X)$);
  else
    $a, b \longleftarrow$ the vertices such that X (b, a)-quasi-dominates T;
    return (`isUkMsparse`$(X, (M \cup \{a\}) \cap X)$ *and* `isMsparse`$(T - X, (M \cup \{b\}) \setminus X)$);

Corollary 1. *The vertex set of a sparse tournament on n vertices can be decomposed into a sequence $U_{n_1}, U_{n_2}, \ldots, U_{n_\ell}$ for some $\ell \leq n$ such that each $T[U_{n_i}]$ dominates or quasi-dominates $T[\bigcup_{i<j\leq\ell} U_{n_j}]$ and $\sum_{i \in [\ell]} n_i = n$.*

5 Degreewidth as a parameter

5.1 Dominating set parameterized by degreewidth

A set of vertices X of a directed graph G is a *dominating set (DS)* if for each vertex $v \in V(G) \setminus X$, we have $N^+(v) \cap X \neq \emptyset$. Observe that in graphs where degreewidth is zero, DS is of size one. Similarly, for tournaments with degreewidth equals to one, the DS is of size at most two. That is, we have trivial solutions for DS for acyclic and sparse tournaments. This motivates us to look for FPT algorithm parameterized by degreewidth. In the following, we develop an FPT algorithm for Dominating Set using universal families. Before that we observe that size of a dominating is always bounded by the size of degreewidth.

Observation 3. *The size of a minimum dominating set of a tournament T is at most $\Delta(T) + 1$.*

Theorem 6. Dominating Set *is FPT in tournaments with respect to degreewidth.*

5.2 FAST and FVST in sparse tournaments

A *forbidden pattern* corresponds to the patterns $\Pi(U_{2k})$ for any $k \geq 1$ as well as $\Pi'(U_4) := \langle v_2, v_4, v_1, v_3 \rangle$. An example of the forbidden pattern $\Pi(U_8)$ is depicted in Figure 4(e). We say a sparse ordering has *forbidden pattern* if a contiguous subsequence of the ordering is a forbidden pattern. Intuitively, the problem of such patterns is that the set of their backward arcs is not a minimum fas. Hopefully, we can use Theorem 4 in such a way that if the pattern $\Pi(U_{2k})$ appears, we can restructure it into $\Pi_{1,2k}(U_{2k})$.

If a sparse ordering does not contain a forbidden pattern then its set of backward arcs is a fas. Hence, we obtain the following result.

Theorem 7. *FAST is solvable in time $O(n^3)$ in sparse tournaments on n vertices.*
For FVST, we show that the problem is difficult to solve on sparse tournaments.

Theorem 8. *FVST is NP-complete on sparse tournaments.*

6 Conclusion

In this paper, we studied a new parameter for tournaments, called degreewidth. We showed that it is NP-hard to decide if degreewidth is at most k, for some natural number k and we proceeded to design a 3-approximation for the degreewidth. One may ask if there is a PTAS for this problem. Then, we investigated sparse tournaments, *i.e.*, tournaments with degreewidth one and developed a polynomial time algorithm to compute a sparse ordering. Is it possible to generalise this result by providing an FPT algorithm to compute the degreewidth? We also showed that FAST can be solved in polynomial time in sparse tournaments, matching with the known result that Arc-Disjoint Triangles Packing and Arc-Disjoint Cycle Packing are both polynomial in sparse tournaments [7]. Therefore, the question arise: can this parameter be used to provide an FPT algorithm for FAST in the general case? Furthermore, we showed an FPT algorithm for DS w.r.t degreewidth. Are there other domination problems e.g., perfect code, partial dominating set, or connected dominating set that is FPT w.r.t degreewidth? Lastly, we also can wonder if this parameter is useful for general digraphs.

Acknowledgements

We would like to thank Frédéric Havet for pointing us a counter-example to the polynomial running-time algorithm in [24, Lemma 35.1, p.97].

References

1. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed parameter algorithms for dominating set and related problems on planar graphs. Algorithmica **33**(4), 461–493 (2002)
2. Allesina, S., Levine, J.M.: A competitive network theory of species diversity. Proceedings of the National Academy of Sciences **108**(14), 5638–5642 (2011)
3. Alon, N.: Ranking tournaments. SIAM J. Discret. Math. **20**(1), 137–142 (2006). https://doi.org/10.1137/050623905, `https://doi.org/10.1137/050623905`
4. Bang-Jensen, J., Gutin, G.Z.: Digraphs - Theory, Algorithms and Applications, Second Edition. Springer Monographs in Mathematics, Springer (2009)
5. Bar-Yehuda, R., Geiger, D., Naor, J., Roth, R.M.: Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and bayesian inference. SIAM J. Comput. **27**(4), 942–959 (1998). https://doi.org/10.1137/S0097539796305109, `https://doi.org/10.1137/S0097539796305109`

6. Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of MAX-3SAT. Electron. Colloquium Comput. Complex. (049) (2003), `http://eccc.hpi-web.de/eccc-reports/2003/TR03-049/index.html`
7. Bessy, S., Bougeret, M., Krithika, R., Sahu, A., Saurabh, S., Thiebaut, J., Zehavi, M.: Packing arc-disjoint cycles in tournaments. In: Rossmanith, P., Heggernes, P., Katoen, J. (eds.) 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany. LIPIcs, vol. 138, pp. 27:1–27:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.MFCS.2019.27, `https://doi.org/10.4230/LIPIcs.MFCS.2019.27`
8. Bessy, S., Bougeret, M., Thiebaut, J.: Triangle packing in (sparse) tournaments: Approximation and kernelization. In: Pruhs, K., Sohler, C. (eds.) 25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria. LIPIcs, vol. 87, pp. 14:1–14:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017). https://doi.org/10.4230/LIPIcs.ESA.2017.14, `https://doi.org/10.4230/LIPIcs.ESA.2017.14`
9. Brandt, F., Fischer, F.: Pagerank as a weak tournament solution. In: Internet and Network Economics: Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007. Proceedings 3. pp. 300–305. Springer (2007)
10. Charbit, P., Thomassé, S., Yeo, A.: The minimum feedback arc set problem is np-hard for tournaments. Comb. Probab. Comput. **16**(1), 1–4 (2007). https://doi.org/10.1017/S0963548306007887, `https://doi.org/10.1017/S0963548306007887`
11. Chen, J., Liu, Y., Lu, S., O'sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 177–186 (2008)
12. Davot, T., Isenmann, L., Roy, S., Thiebaut, J.: Degreewidth: a new parameter for solving problems on tournaments. CoRR **abs/2212.06007** (2022). https://doi.org/10.48550/arXiv.2212.06007, `https://doi.org/10.48550/arXiv.2212.06007`
13. Dechter, R.: Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition. Artif. Intell. **41**(3), 273–312 (1990). https://doi.org/10.1016/0004-3702(90)90046-3, `https://doi.org/10.1016/0004-3702(90)90046-3`
14. Downey, R.G., Fellows, M.R.: Parameterized computational feasibility. In: Feasible mathematics II, pp. 219–244. Springer (1995)
15. Feige, U.: Faster fast(feedback arc set in tournaments). CoRR **abs/0911.5094** (2009), `http://arxiv.org/abs/0911.5094`
16. Fradkin, A.O.: Forbidden Structures and Algorithms in Graphs and Digraphs. Ph.D. thesis, USA (2011), aAI3463323
17. Gavril, F.: " some np-complete problems on graphs", proc. 11th conf. on information sciences and systems, johns hopkins university, baltimore, md (1977)
18. Gurski, F., Rehs, C.: Comparing linear width parameters for directed graphs. Theory Comput. Syst. **63**(6), 1358–1387 (2019). https://doi.org/10.1007/s00224-019-09919-x, `https://doi.org/10.1007/s00224-019-09919-x`
19. Johnson, D.B.: Finding all the elementary circuits of a directed graph. SIAM J. Comput. **4**(1), 77–84 (1975). https://doi.org/10.1137/0204007, `https://doi.org/10.1137/0204007`
20. Karpinski, M., Schudy, W.: Faster algorithms for feedback arc set tournament, kemeny rank aggregation and betweenness tournament. In: Cheong, O.,

Chwa, K., Park, K. (eds.) Algorithms and Computation - 21st International Symposium, ISAAC 2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I. Lecture Notes in Computer Science, vol. 6506, pp. 3–14. Springer (2010). https://doi.org/10.1007/978-3-642-17517-6_3, https://doi.org/10.1007/978-3-642-17517-6_3

21. Kenyon-Mathieu, C., Schudy, W.: How to rank with few errors. In: Johnson, D.S., Feige, U. (eds.) Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007. pp. 95–103. ACM (2007). https://doi.org/10.1145/1250790.1250806, https://doi.org/10.1145/1250790.1250806
22. Laslier, J.F.: Tournament solutions and majority voting, vol. 7. Springer (1997)
23. Leighton, T., Rao, S.: Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. Journal of the ACM (JACM) **46**(6), 787–832 (1999)
24. Thiebaut, J.: Algorithmic and structural results on directed cycles in dense digraphs. (Résultats algorithmiques et structurels sur les cycles orientés dans les digraphes denses). Ph.D. thesis, University of Montpellier, France (2019), https://tel.archives-ouvertes.fr/tel-02491420
25. van Zuylen, A., Williamson, D.P.: Deterministic pivoting algorithms for constrained ranking and clustering problems. Math. Oper. Res. **34**(3), 594–620 (2009). https://doi.org/10.1287/moor.1090.0385, https://doi.org/10.1287/moor.1090.0385