A 7/3-Approximation for Feedback Vertex Sets in Tournaments*

Matthias Mnich†1, **Virginia Vassilevska Williams**‡2, and **László A. Végh**§3

1 **Universität Bonn, Bonn, Germany**
`mmnich@uni-bonn.de`
2 **Computer Science Department, Stanford University, Stanford, USA**
`virgi@cs.stanford.edu`
3 **London School of Economics, London, UK**
`l.vegh@lse.ac.uk`

Abstract

We consider the minimum-weight feedback vertex set problem in tournaments: given a tournament with non-negative vertex weights, remove a minimum-weight set of vertices that intersects all cycles. This problem is NP-hard to solve exactly, and Unique Games-hard to approximate by a factor better than 2. We present the first 7/3 approximation algorithm for this problem, improving on the previously best known ratio 5/2 given by Cai et al. [FOCS 1998, SICOMP 2001].

1998 ACM Subject Classification F2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Approximation algorithms, feedback vertex sets, tournaments

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.67

1 Introduction

Among the most basic concepts in graph theory is the notion of a *feedback vertex set (FVS)* of a digraph: a subset of the vertices S such that removing S makes the digraph acyclic. The computational problem of finding a FVS of minimum size is known as the Feedback Vertex Set problem. A fundamental problem with numerous applications (e.g., in deadlock recovery in operating systems), the Feedback Vertex Set problem is among Karp's 21 original NP-complete problems [13]. Karp's proof of NP-hardness also implies that the problem is APX-hard. Obtaining a constant factor polynomial-time approximation algorithm for the Feedback Vertex Set problem seems elusive and is a major open problem. The best known approximation factor achievable in polynomial time is $O(\log n \log \log n)$ [8, 21].

The Feedback Vertex Set problem is particularly interesting for the special case when the input graph is a *tournament*, i.e., an orientation of the complete graph. The problem restricted to tournaments has many interesting applications, most notably in social choice theory where it is essential to the definition of a certain type of election winners called the Banks set [1].

* Part of this work was done during the trimester program on Combinatorial Optimization at the Hausdorff Institute for Mathematics in Bonn; we would like to thank HIM for their hospitality during the program.
† Supported by ERC Starting Grant 306465 (BeyondWorstCase).
‡ Supported by NSF Grants CCF-1417238, CCF-1528078 and CCF-1514339, and BSF Grant BSF:2012338.
§ Supported by EPSRC First Grant EP/M02797X/1.

24th Annual European Symposium on Algorithms (ESA 2016).
Editors: Piotr Sankowski and Christos Zaroliagis; Article No. 67; pp. 67:1–67:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The Feedback Vertex Set problem remains NP-complete and APX-hard in tournaments. Moreover, Speckenmeyer [22] gave an approximation-ratio preserving polynomial time reduction from the Vertex Cover problem in general undirected graphs to the Feedback Vertex Set problem in tournaments. Consequently, the FVS problem in tournaments cannot be approximated in polynomial time within a factor better than 1.3606, unless $\mathsf{P} = \mathsf{NP}$ [6], and not within a factor better than 2 assuming the Unique Games Conjecture (UGC) [14].

On the upper bound side, the Feedback Vertex Set problem in tournaments admits an easy 3-approximation algorithm: while the tournament contains a directed triangle, place all the triangle vertices in the FVS and remove them from the tournament (see also Bar-Yehuda and Rawitz [2] for another simple 3-approximation algorithm). Cai, Deng and Zang [4] improved the simple algorithm and gave a polynomial time algorithm with approximation guarantee 5/2, even in the case when vertices have non-negative weights and one seeks a solution of approximate minimum weight.

In this paper we develop a 7/3-approximation algorithm for the minimum weight Feedback Vertex Set problem in tournaments, obtaining the first improvement over the eighteen year old result of Cai et al. [4]. Our result shows that the 2.5-approximation ratio is not best possible, and gives hope that a 2-approximation algorithm, that would be optimal under the UGC, might be achievable.

▶ **Theorem 1.** *There exists a polynomial-time* 7/3*-approximation algorithm for finding a minimum-weight feedback vertex set in a tournament.*

In the process of proving the above theorem, we uncover a structural theorem about tournament graphs that has interesting connections to the tournament colouring problem investigated by Berger et al. [3]. We explain these connections in Sect. 5.

1.1 Overview

Let us first give an overview of Cai et al.'s result [4]. Let $\mathcal{T}_5$ denote the set of tournaments on 5 vertices where the minimum FVS has size 2 (note that every tournament on 5 vertices has a FVS of size at most 2). Cai et al. showed that for any tournament free of subtournaments from $\mathcal{T}_5$, the minimum-weight FVS problem becomes polynomial-time solvable. They in fact show that the natural LP relaxation of the problem is integral in $\mathcal{T}_5$-free tournaments: the minimum weight of a FVS equals the maximum value of a fractional directed triangle packing.

For the special case of unit weights only, their 5/2-approximation algorithm starts by greedily choosing subtournaments in $\mathcal{T}_5$, and including all 5 vertices in the FVS. Once the remaining tournament admits no more subtournaments in $\mathcal{T}_5$, the optimal covering algorithm is used. The algorithm returns a 5/2-approximate solution, since every step of removing a subtournament decreases the optimum value by at least 2, and includes 5 vertices in the FVS. The algorithm extends to non-negative weights using the local ratio technique.

We now give an overview of our approach. We define the set $\mathcal{T}_7$ as the set of 7-vertex tournaments where the minimum size of a FVS is 3. The algorithm comprises two stages. The first stage uses the iterative rounding technique, and removes all subtournaments in $\mathcal{T}_7$; the weight of the vertices included at this stage will be at most 7/3-times the decrease in the optimum weight. In the second stage, we give a 7/3-approximate combinatorial algorithm for the remaining $\mathcal{T}_7$-free tournament.

The analogous first stage of Cai et al. obtains a worse factor $5/2$. In the second stage, their algorithm delivers an optimal solution. In contrast, we only give an approximation algorithm in the second stage, but that is sufficient for the overall approximation guarantee.

We now provide some more detail of the two stages. In the first stage we use the iterative rounding technique. We formulate the natural LP relaxation of the minimum-weight FVS problem in the given tournament T, including a covering constraint for every directed triangle of T, and further we include that every subtournament of T belonging to $\mathcal{T}_7$ must be covered by at least three vertices. We consider an optimal solution of the LP relaxation. If there is a vertex of T with fractional value at least $3/7$, we include it in our FVS and remove it from T. We then resolve the LP on the remaining tournament, and again include a vertex with fractional value at least $3/7$, if there exists one. We iterate until there are no more such vertices. At this point, the tournament will be $\mathcal{T}_7$-free, and the fractional optimum value equals exactly one third of the total weight of the vertices (see Lemma 6).

In the second stage, we develop a polynomial time combinatorial algorithm that delivers a FVS of weight at most $7/9$ times the *total weight* of the vertices in a $\mathcal{T}_7$-free tournament (Theorem 4). This algorithm implies our main theorem, since an optimal FVS in the remaining $\mathcal{T}_7$-free tournament is of size at least the optimum fractional value, which by the previous paragraph is exactly a third of the total weight of the nodes, which itself is at least $1/3 \cdot 9/7 = 3/7$ of the size of the FVS returned.

To prove Theorem 4, we decompose the vertex set into "layers". Our algorithm divides the vertices into $\mathcal{T}_5$-free layers, while also identifying a certain vertex set S to be included in the FVS right away. Our final FVS will be composed of the initially selected S, every second layer, and the optimal FVS's inside the remaining layers. To obtain these, we use Cai et al.'s algorithm as a subroutine to find an optimal solution on a $\mathcal{T}_5$-free layer. The layering idea is inspired by Cai et al.'s structural analysis of $\mathcal{T}_5$-free tournaments; nevertheless, we use it quite differently.

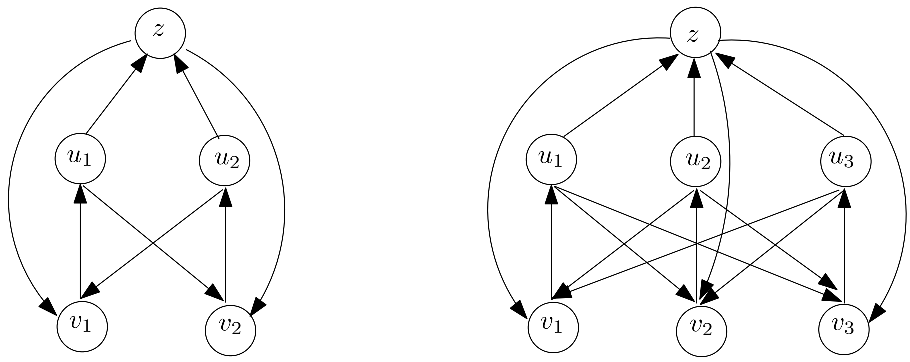
It is natural to conjecture that our approach can be extended to lead to a $(2+\varepsilon)$-approximation for the FVS problem in tournaments, for all $\varepsilon > 0$. At this point it is unclear how to improve the approximation ratio in the above second stage. Nevertheless, our paper provides the next substantial step towards reaching the UGC-based lower bound.

1.2 Related work

Feedback vertex sets in tournaments are a well-studied subject. Dom et al. [7] showed how to decide existence of an FVS of size at most k in time $2^k \cdot n^{O(1)}$, and gave a kernel with $O(k^2)$ vertices. An exponential-time algorithm by Fomin et al. [10] finds an FVS of minimum size in time $O(1.674^n)$, improving on earlier algorithms [18, 7, 11, 16]. Gaspers and Mnich [11] gave a polynomial-space algorithm to enumerate all minimal FVS of a given tournament with polynomial delay; the currently best upper bound on their number is $O(1.6667^n)$ [10].

The related question of FVS in *bipartite* tournaments has also been studied, i.e., orientations of the complete bipartite graph. First, Cai et al. [5] using a similar framework to their $5/2$-approximation algorithm [4], developed a $7/2$-approximation algorithm for FVS in bipartite tournaments. This was improved by Sasatte [20] giving a 3-approximation, and finally, by van Zuylen [23] who developed a polynomial time 2-approximation algorithm.

Iterative rounding is a standard and powerful method in approximation algorithms; we refer the reader to the book by Lau et al. [17]. The approach was made popular by Jain's groundbreaking 2-approximation for survivable network design [12], and the main application area is network design. However, the same principle was already used earlier for various



(a) S_5 **(b)** S_7

Figure 1 Examples of tournaments from $\mathcal{T}_5$ and $\mathcal{T}_7$.

problems. In particular, Krivelevich used implicitly iterative rounding for the undirected triangle cover problem [15]; our application is similar to his argument. Van Zuylen [23] used iterative rounding for FVS in bipartite tournaments.

The Cluster Vertex Deletion problem is another restrictions of the vertex cover problem in 3-uniform hypergraphs. Here the goal is to cover all induced paths of length 2 in an undirected graph. A very recent paper by Fiorini et al. [9] provides a 7/3-approximation algorithm for Cluster Vertex Deletion, improving on the previous best ratio 2.5. An approximation-preserving reduction from the Vertex Cover problem shows that the best possible factor is 2 under the Unique Games Conjecture. Despite these similarities, no approximation-preserving reduction is known between Cluster Vertex Deletion and FVS in Tournaments. The techniques used are also quite different.

2 Description of the Algorithm

Let $T = (V, A)$ be a tournament, equipped with a weight function $w : V \to \mathbb{R}_{\geq 0}$. An arc between $u, v \in V$ will be denoted by $(u, v) \in A$ or $u \to v$. The tournament T is *transitive* if it does not contain any directed cycles, or equivalently, its vertices admit a topological order. A vertex set $S \subseteq V$ is a *feedback vertex set* if $T[V \setminus S]$ is transitive. For a vertex set $S \subseteq V$, let $T - S$ denote the tournament resulting from the removal of the vertex set S from T. If $S = \{v\}$ has a single element, we also use the notation $T - v$.

The following straightforward characterization of FVS's in tournaments is well-known.

▶ **Proposition 2.** *For any tournament T, a set S is a feedback vertex set for T if and only if S intersects every directed triangle of T.*

Let $\mathcal{T}_5$ denote the family of tournaments T' on 5 vertices that do not contain a transitive subtournament on 4 vertices; equivalently, every FVS of T' has size at least 2. The set $\mathcal{T}_5$ contains 3 tournaments, the same ones used by Cai et al. [4]. Characterizations of many related classes of tournaments were given by Sanchez-Flores [19].

Our main focus will be the set $\mathcal{T}_7$ defined as follows. Let $\mathcal{T}_7$ denote the family of tournaments on 7 vertices that do not contain a transitive subtournament on 5 vertices. This is equivalent to the property that every FVS is of size at least 3. We remark that $\mathcal{T}_7$ consists of 121 tournaments.

Fig. 1 gives important examples of tournaments $S_5 \in \mathcal{T}_5$ and $S_7 \in \mathcal{T}_7$. The arcs not included in the figures can be oriented arbitrarily; hence both figures represent multiple

Algorithm 1 Tournament FVS

```
Input: A tournament T = (V, A) with weight function w : V → Q_{≥0}.
Output: A feedback vertex set of T of weight at most (7/3) OPT(T).
Initialize F = ∅, T' = T.
Find an optimal solution x* to (LP).
while T' ≠ ∅ and there exists a vertex v ∈ V(T') with x*_v ≥ 3/7 do
    Set F := F ∪ {v : x*_v ≥ 3/7} and T' := T' \ {v : x*_v ≥ 3/7}.
    Remove every vertex from T' not contained in any directed triangle; denote this
    resulting tournament also by T'.
    Solve (LP) for T' to obtain an optimal solution x*.
If T' ≠ ∅ then run Algorithm Layers (Algorithm 2) for T', returning a FVS F' of T'.
return F ∪ F'.
```

tournaments. Tournament S_5 is identical to F_1 of Cai et al. [4]. We leave the proof of the following simple claim to the reader.

▶ **Proposition 3.** *For the tournaments in Fig. 1, $S_5 \in \mathcal{T}_5$ and $S_7 \in \mathcal{T}_7$.*

For a tournament T, let $\Delta(T)$ denote the family of vertex sets of directed triangles in T. According to Proposition 2, T is transitive if and only if $\Delta(T) = \emptyset$. Similarly, $\mathcal{T}_5(T)$ and $\mathcal{T}_7(T)$ denote the family of vertex sets of the subtournaments of T isomorphic to a tournament in $\mathcal{T}_5$ and $\mathcal{T}_7$, respectively. We say that T is $\mathcal{T}_5$*-free* if $\mathcal{T}_5(T) = \emptyset$ and $\mathcal{T}_7$*-free* if $\mathcal{T}_7(T) = \emptyset$.

We use iterative rounding for the following LP relaxation of the FVS problem in a tournament $T = (V, A)$ with weight function $w : V \to \mathbb{Q}_{\geq 0}$. For a vector $x : V \to \mathbb{R}$ and a set $S \subseteq V$, let $x(S) = \sum_{v \in S} x_v$.

$$\begin{aligned} &\min\ w^T x \\ &x(R) \geq 1 \quad \forall R \in \Delta(T) \\ &x(Q) \geq 3 \quad \forall Q \in \mathcal{T}_7(T) \\ &\quad x \geq 0 \end{aligned} \tag{LP}$$

Notice that (LP) does not impose any constraints for subtournaments in $\mathcal{T}_5(T)$. This is an LP of polynomial size. Let $OPT(T)$ denote the optimum value of (LP).

Our algorithm (Algorithm 1), iteratively builds a FVS F of T, initialized empty. We find an optimal solution x^* to (LP), and as long as there exist vertices v such that $x^*_v \geq \frac{3}{7}$, we include all of them in F and remove them from T. We iterate this process, by resolving the LP for the smaller tournament T'. By the first stage of the algorithm we mean the sequence of these iterative rounding steps, which terminate once T' becomes empty (in which case we are done), or every fractional value x^*_v satisfies $x^*_v < \frac{3}{7}$.

In this case, the current tournament T' must be $\mathcal{T}_7$-free. Indeed, the constraint on the elements of $\mathcal{T}_7(T')$ guarantees that in every $\mathcal{T}_7$ subtournament at least one element must have fractional value at least 3/7. Note that this is true already after the very first iteration. The analogous task of removing all subtournaments from $\mathcal{T}_5(T')$ is done by Cai et al. [4] using the local ratio technique. As shown by Bar-Yehuda and Rawitz [2], this could also be done via a primal-dual algorithm. The local ratio and primal-dual techniques easily give a 3-approximation for the formulation with triangles only (given as (P) in the next section). However, these do not seem to easily extend for our second goal with the iterative rounding, when we only have triangle constraints left, and we proceed as long as there is a vertex of fractional value at least 3/7.

In the second stage we apply Algorithm Layers (Algorithm 2). That is the algorithm described in the following theorem.

▶ **Theorem 4.** *There is an algorithm that, given any $\mathcal{T}_7$-free tournament $T' = (V', A')$ with weight function $w : V' \to \mathbb{Q}_{\geq 0}$, in polynomial time finds a FVS F' of T' of weight at most $\frac{7}{9}w(V')$.*

We defer the description of Algorithm Layers as well as the proof of Theorem 4 to Sect. 4. We now prove the validity of Algorithm 1, provided this result.

3 Proof of Theorem 1

It is straightforward to see that the set $F \cup F'$ returned by the algorithm is a FVS of T. The next simple lemma shows that in every iterative rounding step, the weight of the elements added to F can be bounded by the decrease of $OPT(T)$.

▶ **Lemma 5.** *In every iteration during the first stage of the algorithm with current tournament T' and set F, we have*

$$w(F) \leq \frac{7}{3}(OPT(T) - OPT(T')) \ .$$

Proof. We prove the claim by induction. It is clearly true at the beginning when $T' = T$. Whenever we remove a vertex not contained in any triangle, the left-hand side remains unchanged and the right-hand side may only increase. It is sufficient to prove that if x^* is an optimal solution to (LP) for T' and $S = \{v : x^*_v \geq \frac{3}{7}\} \neq \emptyset$, then $OPT(T' \setminus S) + \frac{3}{7}w(S) \leq OPT(T')$.

Note that x^* restricted to $T' \setminus S$ is feasible to (LP) for $T' \setminus S$, and thus $OPT(T' \setminus S) \leq OPT(T') - \sum_{v \in S} w(v)x^*_v \leq OPT(T') - \frac{3}{7}w(S)$, as required. ◀

As observed above, the tournament T' at the end of the first stage is $\mathcal{T}_7$-free. Theorem 4 guarantees that the FVS F' of T' returned by Algorithm Layers has weight $w(F') \leq \frac{7}{9}w(T')$.

▶ **Lemma 6.** *At the end of the first stage, $OPT(T') = \frac{1}{3}w(T')$.*

Before proving this lemma, let us see how it concludes the proof of Theorem 1. According to Theorem 4 and Lemma 6, $w(F') \leq \frac{7}{9}w(T') \leq \frac{7}{3}OPT(T')$. Using Lemma 5, we see that the weight of the constructed FVS $F \cup F'$ is $w(F \cup F') \leq \frac{7}{3}OPT(T)$.

The proof of Lemma 6 analyzes the LP relaxation with triangle constraints only. At the end of the first stage, T' is $\mathcal{T}_7$-free. Hence, the second set of constraints in (LP) for T' is empty. Let us omit these constraints and write (LP) together with its dual:

$$\begin{aligned} \min \quad & w^T x \\ & x(R) \geq 1 \quad \forall R \in \Delta(T') \\ & x : V \to \mathbb{R}_+ \end{aligned} \qquad \text{(P)} \qquad\qquad \begin{aligned} & \max \mathbf{1}^T y \\ & \sum_{R: v \in R} y_R \leq w_v \quad \forall v \in V' \\ & y : \Delta(T') \to \mathbb{R}_+ \end{aligned} \qquad \text{(D)}$$

Proof of Lemma 6. If $\Delta(T') \neq \emptyset$, then T' is empty and the statement of the lemma holds. Therefore, we can assume that $\Delta(T') \neq \emptyset$, and that $x^*_v \leq \frac{3}{7}$ for every $v \in V'$, where $V' = V(T')$ is the vertex set of T'.

▶ **Claim 7.** *$x^*_v > 0$ for every $v \in V'$.*

Proof. For sake of contradiction, suppose that $x_v^* = 0$ for some $v \in V'$. Every vertex in T' is contained in a directed triangle; say $\{v, u, z\} \in \Delta(T')$. The relaxation (LP) includes a constraint $x_v^* + x_u^* + x_z^* \geq 1$, and therefore $x_u^* \geq \frac{1}{2}$ or $x_z^* \geq \frac{1}{2}$, a contradiction to $x_v^* \leq \frac{3}{7}$ for all $v \in V'$. ◀

By primal-dual slackness, we must have $\sum_{u \in R} y_R = w(u)$ for all $u \in V'$. Then

$$w(V') = \sum_{u \in V'} \sum_{R: u \in R} y_R = \sum_{R \in \Delta(T')} y_R \sum_{u \in R} 1 = 3 \sum_{R \in \Delta(T')} y_R = 3 \cdot OPT(T'),$$

completing the proof. In the third equation, we used that every triangle contains exactly three vertices. ◀

4 The Algorithm Layers

In this section, we present Algorithm Layers and prove Theorem 4. First, we need the following result by Cai et al. [4, Sect. 4].

▶ **Theorem 8** ([4]). *There exists an algorithm that, given any $\mathcal{T}_5$-free tournament $\hat{T}$ with non-negative vertex weights, finds in polynomial time a minimum weight FVS in $\hat{T}$.*

We shall refer to the algorithm as the Cai-Deng-Zang algorithm. We also need a property of $\mathcal{T}_5$-free tournaments established by Cai et al. [4, Thm. 3.2].

▶ **Proposition 9** ([4]). *For any $\mathcal{T}_5$-free tournament $\hat{T}$ with non-negative vertex weights, the minimum weight of a FVS equals the maximum value of a fractional triangle packing.*

Observe that computing the maximum value of a fractional triangle packing amounts to solving (D) to optimality.

The next simple lemma bounds the cost of the FVS found by the Cai-Deng-Zang algorithm in terms of the total weight of the vertices $w(\hat{V})$.

▶ **Lemma 10.** *Let $\hat{T} = (\hat{V}, \hat{A})$ be a $\mathcal{T}_5$-free tournament with weight function $w : \hat{V} \to \mathbb{Q}_{\geq 0}$, and let $\hat{F}$ be an FVS of $\hat{T}$ returned by the* Cai-Deng-Zang *algorithm applied to $(\hat{T}, w)$. Then $w(\hat{F}) \leq w(\hat{V})/3$.*

Proof. By Proposition 9, the polyhedron (P) applied to $T' = \hat{T}$, and w is integral. Setting $x_v = \frac{1}{3}$ for every $v \in \hat{V}$ is a feasible solution, and hence $w(\hat{F}) \leq w(\hat{V})/3$. ◀

4.1 Layers from a vertex

Recall that Theorem 4 takes as input a $\mathcal{T}_7$-free tournament $T' = (V', A')$ with weight function $w : V' \to \mathbb{Q}_{\geq 0}$. For a set $S \subseteq V'$, let $N(S) = \{v \notin S \mid \exists u \in S, v \to u\}$ denote the set of its in-neighbours; let $N(u) := N(\{u\}) = \{v \mid v \to u\}$.

For any vertex $z \in V'$ and $\ell \in \{1, \dots, n\}$, let us define $V_\ell(z)$ as the set of vertices v such that the shortest directed path from v to z has length exactly $\ell - 1$. Equivalently, let $V_1(z) = \{z\}$, $V_2(z) = N(z)$, and for $\ell \geq 2$ let

$$V_{\ell+1}(z) := \{v \in V' \setminus (V_1(z) \cup \ldots \cup V_\ell(z)) \mid \exists u \in V_\ell(z), v \to u\} .$$

These correspond to the layers of the BFS algorithm starting from z. We will prove the following structural result. For two disjoint sets $S, Z \subseteq V'$, let us say that Z *in-dominates* S if for every $s \in S$ there exists a $z \in Z$ with $s \to z$. We say that Z *2-in-dominates* S if Z has a subset $Z' \subseteq Z$ with $|Z'| \leq 2$ such that Z' in-dominates S.

▶ **Theorem 11.** *For every vertex z of positive in-degree, the following hold:*
(a) *The set $V_3(z)$ is $\mathcal{T}_5$-free, and is 2-in-dominated by $V_2(z)$.*
(b) *The set $V_4(z)$ is $\mathcal{T}_5$-free, and is 2-in-dominated by $V_3(z)$.*
(c) *If z is a minimum in-degree vertex in the tournament, then $V_3(z) \neq \emptyset$, and $V_2(z)$ is also $\mathcal{T}_5$-free.*

The proof of Theorem 11 is given in Sect. 4.4. Let us now provide some context and motivation. Cai et al. [4] showed that for any $\mathcal{T}_5$-free tournament, if we select a minimum in-degree vertex z, then every layer $V_i(z)$ induces a transitive tournament and is 1-in-dominated by $V_{i-1}(z)$. This is an important step in their algorithm for finding the exact optimal solution in $\mathcal{T}_5$-free tournaments.

Assume that the analogous property held for $\mathcal{T}_7$-free tournaments T': starting from a minimum in-degree vertex z, every layer $V_{i-1}(z)$ is $\mathcal{T}_5$-free. Then one could get a FVS of T' with weight at most $\frac{2}{3}w(V')$ as follows. Compare the total weight of the even and odd layers, and include in the FVS whichever of the two is smaller. Let us assume the total weight of the odd layers is smaller; the argument is same for the other case. For every remaining even layer $V_i(z)$, run the Cai-Deng-Zang algorithm to obtain a FVS F_i of $V_i(z)$. Form the final FVS F' of T' as the union of all odd layers and the union of the F_i's for the even layers. Using Proposition 10, it is easy to verify that $w(F') \leq \frac{2}{3}w(V')$. Further, F' will be a FVS of T', since by the construction of the $V_i(z)$'s, every triangle must fall on consecutive layers. That is, it is, if a triangle T intersects layers $V_i(z)$ and $V_j(z)$ with $i < j$, then $j \leq i+2$, and if $j = i+2$ then T must also intersect layer $V_{i+1}(z)$.

However, Theorem 11 only claims $\mathcal{T}_5$-freeness of layers $V_i(z)$ for $i \leq 4$. This property might not hold for higher values of i. To overcome this difficulty, we modify the layering procedure. While the layers are constructed, we already include certain vertices in the final FVS. This is to make sure that for every layer U_i, it holds that $U_i = V_j(z')$ in some subtournament of T, for a certain vertex z' in a previous layer and $j = 3$ or $j = 4$. Hence Theorem 11 guarantees that all constructed layers are $\mathcal{T}_5$-free. The construction of the final FVS will be a modification of the simple argument above.

4.2 Description of the layering algorithm

The algorithm (Algorithm 2) first partitions the vertex set V' into $S \cup \bigcup_{j=1}^{2k} U_j$ for some $2k \leq n$. We now describe how the layers are constructed in Steps 1-11. We start by setting $U_1 = \{z_1\}$ for a vertex z_1 of minimum in-degree. We let $U_2 = N(z_1)$ be the set of in-neighbours of z_1. The set W will denote the set of vertices not yet included in some U_k or in S; at this point, $W = V' \setminus (U_1 \cup U_2)$.

While W is not empty, we construct an odd layer U_{2k+1}, an even layer U_{2k+2}, and S_{2k+1} as follows. First consider the case when U_{2k} has at least one in-neighbour in W. We set $U_{2k+1} = N(U_{2k}) \cap W$, and remove U_{2k+1} from W. Let U' be the set of in-neighbours of U_{2k+1} in W. We note that $U' = \emptyset$ is possible. We partition U' into U_{2k+2} and S_{2k+2}, and remove U' from W. To obtain this partitioning, we pick a vertex $z_{2k+1} \in U_{2k+1}$ such that $w(N(z_{2k+1}) \cap U') \geq w(U')/2$. The existence of such a vertex z_{2k+1} is non-trivial, and will be proved in Lemma 12(c). We set $U_{2k+2} = N(z_{2k+1}) \cap U'$, and $S_{2k+2} = U' \setminus U_{2k+2}$; the set S_{2k+2} will be part of S.

Let us now address the case when U_{2k} does not have any in-neighbours in W. In this case, we select $U_{2k+1} = \{z\}$ for a vertex $z \in W$ that has minimum in-degree inside W. We refer to the latter scenario as a *fresh start.* We set U_{2k+2} as the set of in-neighbours of z in W, and remove these vertices from W; here $U_{2k+2} = \emptyset$ is possible.

Algorithm 2 LAYERS

Input: A $\mathcal{T}_7$-free tournament $T' = (V', A')$ with weight function $w : V' \to \mathbb{Q}_{\geq 0}$.
Output: A feedback vertex set F' of T' of weight at most $\frac{7}{9}w(V')$.

1: Choose z_1 as a vertex of minimum in-degree.
2: Set $U_1 := \{z_1\}$,
3: Set $U_2 := N(z_1)$, $W := V' \setminus (U_1 \cup U_2)$, $k := 1$.
4: **while** $W \neq \emptyset$ **do**
5:     **if** $N(U_{2k}) \cap W \neq \emptyset$ **then**
6:         Set $U_{2k+1} := N(U_{2k}) \cap W$,
7:         $W := W \setminus U_{2k}$.
8:         Set $U' := N(U_{2k+1}) \cap W$, $W := W \setminus U'$.
9:         Choose $z_{2k+1} \in U_{2k+1}$ such that $w(U' \cap N(z_{2k+1})) \geq w(U')/2$.
10:         Set $U_{2k+2} := U' \cap N(z_{2k+1})$; $S_{2k+2} := U' \setminus N(z_{2k+1})$.
11:     **else**   *//fresh start*
12:         Choose $z \in W$ with $|N(z) \cap W|$ minimal.
13:         Set $U_{2k+1} := \{z\}$, $U_{2k+2} := N(z) \cap W$, and $S_{2k+2} := \emptyset$.
14:         Set $W := W \setminus (U_{2k+1} \cup U_{2k+2})$.
15:     Set $k := k + 1$.
16: Set $L_0 := \cup_{j=1}^{k} U_{2j}$, $L_1 := \cup_{j=0}^{k-1} U_{2j+1}$, and $S := \cup_{j=1}^{k} S_{2j}$.
17: **if** $w(L_0) \geq w(L_1)$ **then**
18:     Run the CAI-DENG-ZANG algorithm for every U_{2j} to obtain a FVS F_{2j} of U_{2j}.
19:     Set $F' := (\cup_{j=1}^{k} F_{2j}) \cup S \cup L_1$.
20: **else**
21:     Run the CAI-DENG-ZANG algorithm for every U_{2j+1} to obtain a FVS F_{2j+1} of U_{2j+1}.
22:     Set $F' := (\cup_{j=0}^{k-1} F_{2j+1}) \cup S \cup L_0$.
23: **return** F'.

The layering procedure finishes once $W = \emptyset$. At this point, we denote by $L_0 = \bigcup_{j=1}^{k} U_{2j}$ the set of all even and by $L_1 = \bigcup_{j=0}^{k-1} U_{2j+1}$ the set of all odd layers, and by $S = \bigcup_{j=1}^{k} S_{2j}$ the set of vertices removed during the procedure. Thus, $V' = S \cup L_0 \cup L_1$. Given the layering, the algorithm constructs a FVS in Steps 12-18 as follows. If $w(L_0) \geq w(L_1)$, then we use the CAI-DENG-ZANG algorithm to find an optimal FVS F_{2j} in all even layers U_{2j}. We set the entire FVS as $F' := (\cup_{j=1}^{k} F_{2j}) \cup S \cup L_1$. Otherwise, we use the CAI-DENG-ZANG algorithm in all odd layers to find optimal FVS's F_{2j+1}, and set $F' := (\cup_{j=0}^{k-1} F_{2j+1}) \cup S \cup L_0$.

The algorithm clearly runs in polynomial time: every while cycle decreases the size of W by at least one, and every step amounts to examining in-neighbourhoods of vertices and comparing weights of sets.

4.3 Proof of correctness

The following lemma summarizes the essential properties of the layering obtained.

▶ **Lemma 12.** *The sets S and U_i returned by Algorithm* LAYERS *satisfy the following properties.*

(a) *If $i > j + 1$, then $u \to v$ for every $u \in U_j$ and $v \in U_i$.*
(b) *Every subtournament $T'[U_i]$ is $\mathcal{T}_5$-free.*
(c) *There always exists a vertex $z_{2i+1} \in U_{2i+1}$ as required in line 9 of the algorithm.*
(d) *$w(S) \leq w(L_0)$.*

Proof. Part (a) is immediate, since if $u \in U_j$, then $N(u) \subseteq \cup_{\ell=0}^{j+1}(U_\ell \cup S_\ell)$ (let us use the convention $S_\ell = \emptyset$ for all odd values of ℓ).

We prove parts (b) and (c) simultaneously. We prove it for all layers before the first fresh start happens. If $N(z_1) = \emptyset$, then $U_2 = \emptyset$, hence U_1 and U_2 are trivially $\mathcal{T}_5$-free, and U_3 will be obtained by a fresh start. Otherwise, part (b) is a direct consequence of Theorem 11 for layers $1 \le i \le 4$, as z_1 was chosen as a minimum in-degree vertex; note that $V_3(z_1) \neq \emptyset$ and hence $U_3 = V_3(z_1)$ was not obtained by a fresh start. In this case, the existence of vertex $z_3 \in U_3$ follows by Theorem 11(c): $U' = V_4(z_1)$, and thus U' is 2-in-dominated by U_3. This means that there exist $z, z' \in U_3$ such that $N(z) \cup N(z') \supseteq U'$ (we allow $z = z'$). Without loss of generality, we may assume $w(U' \cap N(z)) \ge w(U' \cap N(z'))$. Then $z_3 = z$ gives an appropriate choice.

Assuming that U_5 is not obtained by a fresh start, let us apply Theorem 11 in the tournament T'' that is the restriction of T' to the ground set $\{z_3\} \cup U_4 \cup U_5 \cup (U_6 \cup S_6)$. In T'' we have $V_3(z_3) = U_5$ and $V_4(z_3) = U_6 \cup S_6$, and therefore U_5 and $U_6 \cup S_6$ are both $\mathcal{T}_5$-free. Further, $U_6 \cup S_6$ is 2-in-dominated by U_5 and therefore we can choose an appropriate $z_5 \in U_5$ as above. The same argument works for all values of $i \ge 3$: consider the restriction of T' to $\{z_{2i-1}\} \cup U_{2i} \cup U_{2i+1} \cup (U_{2i+2} \cup S_{2i+2}$, and apply Theorem 11. We obtain that U_{2i+1} and $U_{2i+2} \cup S_{2i+2}$ are $\mathcal{T}_5$-free as well as the choice of $z_{2i+1} \in U_{2i+1}$.

Assume now that a certain layer U_{2i+1} is obtained by a fresh start. Then we can apply the same argument as above to show parts (b) and (c) for all subsequent layers until the next fresh start: we restrict the tournament from V to the ground set W at the beginning of the iteration when U_{2i+1} is constructed.

Finally, part (d) is straightforward, since $w(U_{2i+2}) \ge w(S_{2i+2})$ by the choice of z_{2i+2}. ◀

We are ready to prove the correctness and approximation ratio of the algorithm.

Proof of Theorem 4. By Lemma 12(b), the Cai-Deng-Zang algorithm can be applied in all layers U_i and finds an optimal FVS F_i in polynomial time.

First, let us show that the set F' returned by Algorithm Layers is indeed a FVS of T'. For a contradiction, assume $V' \setminus F'$ contains a directed triangle uvs.

Let us assume $w(L_0) \ge w(L_1)$; the other case follows similarly. In this case, $V' \setminus F' \subseteq L_0$. The three vertices u, v and s cannot fall into the same layer U_{2i}, as in every such layer we removed a FVS F_{2i}. Hence they must fall into at least two different U_{2i}'s. By Lemma 12(a), if vertices fall into different even layers, then all arcs from the lower layers point towards the higher layers, excluding the possibility of such a triangle.

The proof is complete by showing $w(F') \le \frac{7}{9}w(V')$, or equivalently, $w(V' \setminus F') \ge \frac{2}{9}w(V')$.

Case I: $w(L_0) \ge w(L_1)$. In this case, $w(V' \setminus F') = \cup_{j=1}^{k}(U_{2j} \setminus F_{2j})$. By Proposition 10, $w(F_{2j}) \le w(U_{2j})/3$ for all layers, and thus $w(V' \setminus F') \ge \frac{2}{3}w(L_0)$. Using Lemma 12(d), $w(L_0) \ge \max\{w(L_1), w(S)\}$, and thus $w(L_0) \ge w(V')/3$. Thus $w(V' \setminus F') \ge \frac{2}{9}w(V')$ follows.

Case II: $w(L_0) < w(L_1)$. Using the same argument as in the previous case, we obtain $w(V' \setminus F') \ge \frac{2}{3}w(L_1)$. Again using Lemma 12(d), $w(L_1) > w(L_0) \ge w(S)$, and therefore $w(L_1) \ge w(V')/3$, implying $w(V' \setminus F') \ge \frac{2}{9}w(V')$. ◀

4.4 Proof of Theorem 11

Let us first verify part (c):

▶ **Lemma 13.** *Let z be a minimum in-degree vertex in a $\mathcal{T}_7$-free tournament. Then $V_2(z)$ is $\mathcal{T}_5$-free. If $V_2(z) \neq \emptyset$, then $V_3(z) \neq \emptyset$.*

Proof. The claim is trivial if $V_2(z) = \emptyset$. Hence we assume $V_2(z) \neq \emptyset$ in the sequel. We first claim that for every $u \in V_2(z)$ there must exist a $v \in V_3(z)$ with $v \to u$. Indeed, assume that for some u there exists no such v. Then $N(u) \subsetneq V_2(z) = N(z)$ must hold. This is a contradiction to the choice of z with $|N(z)|$ minimum. This already shows that $V_3(z) \neq \emptyset$.

Consider a subset $H \subseteq V_3(z)$ containing at least one vertex v with $v \to u$ for every $u \in V_2$; choose H minimal for containment. If $|H| \geq 3$, then there must be three vertices $v_1, v_2, v_3 \in H$, and three vertices $u_1, u_2, u_3 \in V_2(z)$ such that $v_i \to u_i$ for $i = 1, 2, 3$, while $u_i \to v_j$ if $i \neq j$. Then z and these vertices together form an $S_7 \in \mathcal{T}_7$ subtournament as in Fig. 1(b), a contradiction.

Hence $|H| \leq 2$. For a contradiction, assume $X \subseteq V_2(z)$ forms a $\mathcal{T}_5$-graph ($|X| = 5$). There exists a $v \in H$ with $|\{s \in X : v \to s\}| \geq 3$. We claim that $X \cup \{v, z\} \in \mathcal{T}_7$. Indeed, assume it contains a transitive tournament Y on 5 vertices. Since $X \in \mathcal{T}_5$, $|X \cap Y| \leq 3$; hence $v, z \in Y$ and $|X \cap Y| = 3$. There must be a vertex $t \in X \cap Y$ with $v \to t$, and thus vtz is a directed triangle, a contradiction. ◀

For (a) and (b) of Theorem 11, we show that the 2-in-domination claim implies $\mathcal{T}_5$-freeness:

▶ **Lemma 14.** *Let z be an arbitrary vertex in a $\mathcal{T}_7$-free tournament. For $i \geq 3$, if $V_i(z)$ is 2-in-dominated by $V_{i-1}(z)$, then $V_i(z)$ is $\mathcal{T}_5$-free.*

Proof. Consider a $\mathcal{T}_5$-subtournament X in $V_i(z)$. By 2-in-domination, there must be a $v \in V_{i-1}(z)$ such that $|N(v) \cap X| \geq 3$. Let $s \in V_{i-2}(z)$ be such that $v \to s$. We obtain a contradiction as in the previous proof, showing that $X \cup \{v, s\} \in \mathcal{T}_7$. Indeed, assume that $X \cup \{v, s\}$ has a transitive subtournament Y of size 5. We have $|X \cap Y| \leq 3$ since X is $\mathcal{T}_5$; thus $|X \cap Y| = 3$, and $v, s \in Y$. But then there exists a vertex $t \in X \cap Y \cap N(v)$. We have $s \to t$ because $N(s) \cap V_{i-1}(z) = \emptyset$. Thus stv is a directed triangle. ◀

The proof of Theorem 11 is complete by the following two lemmata, that show that both $V_3(z)$ and $V_4(z)$ are 2-in-dominated by the previous layer.

▶ **Lemma 15.** *For an arbitrary vertex z in a $\mathcal{T}_7$-free tournament T', the set $V_3(z)$ is 2-in-dominated by $V_2(z)$.*

Proof. Let $H \subseteq V_2(z)$ be a minimal set for containment that in-dominates $V_3(z)$. We show that $|H| \leq 2$. Indeed, if $|H| \geq 3$, then again there must be a tournament $S_7 \in \mathcal{T}_7$ as in Fig. 1(b), formed by z, three vertices in $V_2(z)$ and three in $V_3(z)$. ◀

In the sequel, let $\{a, b\} \subseteq V_2(z)$ be a set that 2-in-dominates $V_3(z)$.

▶ **Lemma 16.** *For an arbitrary vertex z in a $\mathcal{T}_7$-free tournament T', the set $V_4(z)$ is 2-in-dominated by $V_3(z)$.*

Proof. For sake of contradiction, assume that any minimal set in $V_3(z)$ that 2-in-dominates $V_4(z)$ has size at least 3. Then there must exists vertices $u_1, u_2, u_3 \in V_3(z)$ and $v_1, v_2, v_3 \in V_4(z)$ such that $v_i \to u_i$ for $i = 1, 2, 3$, while $u_i \to v_j$ if $i \neq j$.

If all $u_1, u_2, u_3 \in N(a)$, then $\{a, u_1, u_2, u_3, v_1, v_2, v_3\}$ forms a tournament in $\mathcal{T}_7$, a contradiction. A similar argument applies for b. We may therefore assume (by possibly renaming the indices) that $u_1 \to a, u_2 \to a, u_3 \to b, a \to u_3$, and $b \to u_2$. See Fig. 2.

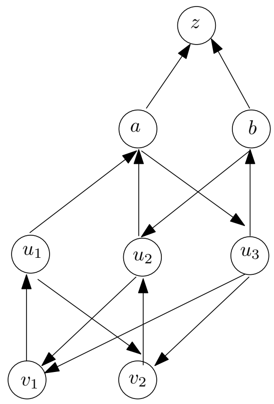


Figure 2 Illustration of the proof of Lemma 16. A few directed edges that are not portrayed are: from z to each one of $\{u_1, u_2, u_3, v_1, v_2\}$ and from each of $\{a, b\}$ to each of $\{v_1, v_2\}$.

Since T' is $\mathcal{T}_7$-free, then every 7-vertex subgraph of $\{z, a, b, u_1, u_2, u_3, v_1, v_2, v_3\}$ must contain a transitive tournament on 5 vertices. Let $Q = \{z, a, b, u_2, u_3\}$, and for $i = 1, 2$, let $Q_i = Q \cup \{u_1, v_i\}$. Let T_i be a transitive tournament on 5 nodes in Q_i.

Notice that Q forms a $\mathcal{T}_5$. Because of this, for $i = 1, 2$ it holds $\{u_1, v_i\} \subseteq T_i$. Furthermore, b, u_3, z cannot all be in T_i since they form a directed triangle; so $\{a, u_2\} \cap T_i \neq \emptyset$. A symmetric argument shows that $\{b, u_3\} \cap T_i \neq \emptyset$ as well.

Now, since either $u_2 \to u_1$ or $u_1 \to u_2$, either $u_2 u_1 v_2$ or $u_1 u_2 v_1$ forms a directed triangle. Thus, $u_2 \notin T_i$ for either $i = 1$ or $i = 2$. For the same i, $a \in T_i$ because of $\{a, u_2\} \cap T_i \neq \emptyset$. Then z cannot be in T_i because $u_1 a z$ forms a directed triangle. Hence $T_i = \{a, b, u_1, u_3, v_i\}$, and this implies that (i) $a \to b$ since $a \to u_3 \to b$, (ii) $u_1 \to u_3$ since $u_1 \to a \to u_3$, and (iii) $u_1 \to b$ since $u_1 \to a \to b$, using (i).

As noted above, $\{u_1, v_1\} \subseteq T_1$. By (ii), $v_1 u_1 u_3$ forms a directed triangle, and by (iii), $v_1 u_1 b$ forms a triangle. Hence, neither u_3 nor b can be contained in T_1, contradicting that $\{b, u_3\} \cap T_1 \neq \emptyset$. This completes the proof of Lemma 16. ◀

5 Connections to Tournament Colouring

We explore a connection to the notion of heroes and celebrities in tournaments studied by Berger, Choromanski, Chudnovsky, Fox, Loebl, Scott, Seymour and Thomassé [3]. Colouring a tournament means partitioning its vertex set into transitive subtournaments; the chromatic number of a tournament is the minimum number of colours needed. A tournament H is called a *hero*, if there exists a constant c_H such that every H-free tournament has chromatic number at most c_H. Further, H is called a *celebrity*, if for some constant $c'_H > 0$, every H-free tournament T has a transitive subtournament of size at least $c'_H |V(T)|$. Clearly, every hero is a celebrity; Berger et al. show that the converse also holds: every celebrity is a hero. Their work gives a characterization of all tournaments that are heros (or equivalently, celebrities).

In this context, our Theorem 4 shows that $\mathcal{T}_7$ collectively form a celebrity set. Further, our constant $c' = 2/9$ seems much better than the constants that could be derived using the techniques of Berger et al. [3]. The set $\mathcal{T}_7$ includes some heros as well as some non-hero tournaments. In contrast, the set $\mathcal{T}_5$ is precisely the set of heros on 5 vertices.

Berger et al.'s [3] characterization rules out the following possible modification of our algorithm to obtain a 2-approximation for the Feedback Vertex Set in tournaments problem. Instead of $\mathcal{T}_7$, one could use the single tournament ST_6, the unique 6-vertex tournament not containing a transitive subtournament of order 4 [19]. All copies ST_6 can be removed from the input tournament by losing a factor 2 in the approximation ratio only (instead of losing 7/3 by removing copies of subtournaments from $\mathcal{T}_7$). However, according to Berger et al. [3, Thm. 1.2], ST_6 is *not* a hero, and hence there is no hope to prove a version of Theorem 4 for this setting.

References

1 Jeffrey S. Banks. Sophisticated voting outcomes and agenda control. *Soc. Choice Welf.*, 1(4):295–306, 1985.

2 Reuven Bar-Yehuda and Dror Rawitz. On the equivalence between the primal-dual schema and the local ratio technique. *SIAM J. Discrete Math.*, 19(3):762–797, 2005.

3 Eli Berger, Krzysztof Choromanski, Maria Chudnovsky, Jacob Fox, Martin Loebl, Alex Scott, Paul Seymour, and Stéphan Thomassé. Tournaments and colouring. *J. Combin. Theory Ser. B*, 103(1):1–20, 2013.

4 Mao-Cheng Cai, Xiaotie Deng, and Wenan Zang. An approximation algorithm for feedback vertex sets in tournaments. *SIAM J. Comput.*, 30(6):1993–2007 (electronic), 2001.

5 Mao-cheng Cai, Xiaotie Deng, and Wenan Zang. A min-max theorem on feedback vertex sets. *Math. Oper. Res.*, 27(2):361–371, 2002.

6 Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Ann. of Math. (2)*, 162(1):439–485, 2005.

7 Michael Dom, Jiong Guo, Falk Hüffner, Rolf Niedermeier, and Anke Truss. Fixed-parameter tractability results for feedback set problems in tournaments. *J. Discrete Algorithms*, 8(1):76–86, 2010.

8 Guy Even, Joseph (Seffi) Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

9 Samuel Fiorini, Gwenaël Joret, and Oliver Schaudt. Improved approximation algorithms for hitting 3-vertex paths. In *Proc. IPCO 2016*, volume 9682 of *Lecture Notes Comput. Sci.*, pages 238–249, 2016.

10 Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, and Saket Saurabh. Exact algorithms via monotone local search. In *Proc. STOC 2016*, pages 764–775, 2016.

11 Serge Gaspers and Matthias Mnich. Feedback vertex sets in tournaments. *J. Graph Theory*, 72(1):72–89, 2013.

12 Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

13 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, 1972.

14 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\epsilon$. *J. Comput. System Sci.*, 74(3):335–349, 2008.

15 Michael Krivelevich. On a conjecture of Tuza about packing and covering of triangles. *Discrete Math.*, 142(1):281–286, 1995.

16 Mithilesh Kumar and Daniel Lokshtanov. Faster exact and parameterized algorithm for feedback vertex set in tournaments. In *Proc. STACS 2016*, volume 47 of *Leibniz Int. Proc. Informatics*, pages 49:1–49:13, 2016.

17 Lap Chi Lau, R. Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization.* Cambridge Texts in Applied Mathematics. Cambridge University Press, New York, 2011.

18 John W. Moon. On maximal transitive subtournaments. *Proc. Edinburgh Math. Soc. (2)*, 17:345–349, 1970/71.

19 Adolfo Sanchez-Flores. On tournaments free of large transitive subtournaments. *Graphs Comb.*, 14(2):181–200, 1998.
20 Prashant Sasatte. Improved approximation algorithm for the feedback set problem in a bipartite tournament. *Oper. Res. Lett.*, 36(5):602–604, 2008.
21 Paul D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.
22 Ewald Speckenmeyer. On feedback problems in digraphs. In *Proc. WG 1989*, volume 411 of *Lecture Notes Comput. Sci.*, pages 218–231. Springer, 1990.
23 Anke van Zuylen. Linear programming based approximation algorithms for feedback set problems in bipartite tournaments. *Theor. Comput. Sci.*, 412(23):2556–2561, 2011.