Discrete Applied Mathematics 337 (2023) 149–160

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

A 7/3-approximation algorithm for feedback vertex set in tournaments via Sherali–Adams[☆]

Manuel Aprile [a,*], Matthew Drescher [b], Samuel Fiorini [c], Tony Huynh [d]

[a] Università degli Studi di Padova, Italy
[b] UC Davis, United States of America
[c] Université libre de Bruxelles, Belgium
[d] Sapienza Università di Roma, Italy

ARTICLE INFO

Article history:
Received 16 June 2021
Received in revised form 4 April 2023
Accepted 13 April 2023
Available online 19 May 2023

Keywords:
Feedback vertex set
Tournaments
Approximation algorithm
Integrality gap
Sherali–Adams hierarchy

ABSTRACT

We study the feedback vertex set problem in tournaments from the polyhedral point of view, and in particular we show that performing just one round of the Sherali–Adams hierarchy gives a relaxation with integrality gap 7/3. This allows us to derive a 7/3-approximation algorithm for the feedback vertex set problem in tournaments that matches the best deterministic approximation guarantee due to Mnich, Williams, and Végh, and is a simplification and runtime improvement of their approach.

1. Introduction

A *feedback vertex set* (FVS) of a tournament T is a set X of vertices such that $T - X$ is acyclic. Given a tournament T and (vertex) weights $w : V(T) \to \mathbb{Q}_{\geq 0}$, the *feedback vertex set problem in tournaments* (FVST) asks to find a feedback vertex set X such that $w(X) := \sum_{x \in X} w(x)$ is minimum. This problem has numerous applications, for example in determining election winners in social choice theory [4].

We let $\mathrm{OPT}(T, w)$ be the minimum weight of a feedback vertex set of the weighted tournament (T, w). An *α-approximation algorithm* for FVST is a polynomial-time algorithm computing a feedback vertex set X with $w(X) \leq \alpha \cdot \mathrm{OPT}(T, w)$. We say that a directed cycle on 3 vertices is a *directed triangle*.

Let T be a tournament and let $\triangle(T)$ denote the collection of all $\{a, b, c\} \subseteq V(T)$ that induce a directed triangle in T; we will use $abc := \{a, b, c\}$ as a convenient shorthand. Note that T is acyclic if and only if $\triangle(T) = \emptyset$. This suggests the following easy 3-approximation algorithm for FVST in the unweighted case (the general case follows for instance from the *local ratio technique* [7]). If T is acyclic, then $\varnothing$ is an FVS, and we are done. Otherwise, we find $abc \in \Delta(T)$ and put all its vertices into the FVS. We then replace T by $T - \{a, b, c\}$ and recurse.

In this paper, we study FVST from the polyhedral perspective. The *basic LP relaxation*, or just basic relaxation, for T is the polytope

$$P(T) := \{x \in [0, 1]^{V(T)} \mid \forall abc \in \triangle(T) : x_a + x_b + x_c \geq 1\},$$

where throughout the text LP stands for linear programming (or linear program, according to the context).

☆ This project was supported by ERC Consolidator Grant 615640-ForEFront. Samuel Fiorini and Manuel Aprile were also supported by FNRS grant T008720F-35293308-BD-OCP. Tony Huynh was also supported by the Australian Research Council.
* Corresponding author.
E-mail address: manuel.aprile@unipd.it (M. Aprile).

https://doi.org/10.1016/j.dam.2023.04.016

The integer points of $P(T)$ are exactly the characteristic vectors of feedback vertex sets of T. We call any polytope $Q(T) \subseteq [0,1]^{V(T)}$ with this property a *relaxation* of FVST. The extent to which optimizing over a relaxation $Q(T)$ of FVST approximates $\mathrm{OPT}(T,w)$ is encoded by its *integrality gap*. This is the supremum of the ratio

$$\frac{\mathrm{OPT}(T,w)}{\min\{w^T x : x \in Q(T)\}},$$

taken over all $w : V(T) \to \mathbb{Q}_{\geq 0}$.

It is not hard to see (Proposition 4) that the integrality gap of $P(T)$ is at most 3. Moreover, using random tournaments, one can show (see Proposition 3) that the worst-case integrality gap of $P(T)$ is exactly 3 (by *worst-case*, we mean that we take the supremum over all tournaments T).

In this paper, we give a relaxation for FVST with integrality gap 7/3. This allows us to derive a 7/3-approximation algorithm for FVST, matching the best-known deterministic approximation algorithm by Mnich et al. [14]. Our approach simplifies the algorithm from [14] and has significantly faster running time. Before stating our results, we briefly review the previous work on the topic.

State of the art

The first non-trivial approximation algorithm for FVST was a 5/2-approximation algorithm by Cai, Deng, and Zang [5]. Cai et al.'s approach is polyhedral. It is based on the fact that the basic LP relaxation $P(T)$ is integral whenever the input tournament avoids certain subtournaments, see the next paragraphs for details.

Let $\mathcal{T}_5$ be the set of tournaments on 5 vertices where the minimum FVS has size 2. Up to isomorphism, $|\mathcal{T}_5| = 3$ (see Cai et al. [5]). We say that T is $\mathcal{T}_5$*-free* if no subtournament of T is isomorphic to a member of $\mathcal{T}_5$. More generally, let $\mathcal{T}$ be a collection of tournaments. A $\mathcal{T}$*-subtournament* of T is a subtournament of T that is isomorphic to some tournament of $\mathcal{T}$. We say that T is $\mathcal{T}$*-free* if T does not contain a $\mathcal{T}$-subtournament.

Cai et al. prove that $P(T)$ is integral as soon as T is $\mathcal{T}_5$-free. In this case solving a polynomial-size LP gives a minimum weight FVS. We let $\mathrm{CDZ}(T,w)$ be the polynomial-time algorithm from Cai et al. [5], that given a $\mathcal{T}_5$-free tournament T and $w : V(T) \to \mathbb{Q}_{\geq 0}$, finds a minimum weight feedback vertex set of T.

A 5/2-approximation algorithm follows directly from this. Using the local ratio technique, while T contains a $\mathcal{T}_5$-subtournament S, one can reduce to a smaller instance with one vertex of S removed. If one is aiming for a 5/2-approximation algorithm, one can reduce to a $\mathcal{T}_5$-free tournament T, for which one can even solve the problem exactly by applying $\mathrm{CDZ}(T,w)$.

The 5/2-approximation algorithm of Cai et al. [5] was improved to a 7/3-approximation algorithm by Mnich, Williams, and Végh [14]. Loosely speaking, Mnich et al.'s algorithm replaces $\mathcal{T}_5$ by $\mathcal{T}_7$, defined as the set of tournaments on 7 vertices where the minimum FVS has size 3. It is known that, up to isomorphism, $|\mathcal{T}_7| = 121$ (see [14]).

Similarly, if one is aiming for a 7/3-approximation algorithm, one can reduce to $\mathcal{T}_7$-free tournaments. In fact, instead of using the local ratio technique, Mnich et al. [14] use iterative rounding, see the next paragraph. However, the basic relaxation $P(T)$ is not necessarily integral for $\mathcal{T}_7$-free tournaments, so obtaining a 7/3-approximation algorithm requires more work.

The algorithm in [14] consists of two phases. Let the $\mathcal{T}_7$*-relaxation* be the LP obtained from the basic relaxation $P(T)$ by adding the constraint $\sum_{v \in V(S)} x_v \geq 3$ for each $\mathcal{T}_7$-subtournament S of T. The first phase is an iterative rounding procedure on the $\mathcal{T}_7$-relaxation. This reduces the problem to a residual tournament which is $\mathcal{T}_7$-free. The second phase is a 7/3-approximation algorithm for FVST on the residual tournament, via an intricate layering procedure.

Recently, Lokshtanov, Misra, Mukherjee, Panolan, Philip, and Saurab [12,13] gave a *randomized* 2-approximation algorithm for FVST. Their algorithm produces a feasible solution that is a 2-approximation with probability at least 1/2, and does not rely on Cai et al. [5], but rather on the idea of guessing vertices which are not part of some optimal FVS and that of controlling the in-degree sequence of the tournament. The derandomized version of their algorithm runs in quasi-polynomial-time. A deterministic 2-approximation algorithm would be best possible, since for every $\varepsilon > 0$, FVST does not have a $(2-\varepsilon)$-approximation algorithm, unless the Unique Games Conjecture is false or $\mathsf{P} = \mathsf{NP}$.[1]

Our contribution

We show that performing just one round of the *Sherali–Adams hierarchy* [16] on the basic relaxation $P(T)$ yields a relaxation whose integrality gap is 7/3. Below, $\mathsf{SA}_r(T,w)$ denotes both the relaxation obtained by performing r rounds of the Sherali–Adams hierarchy on $P(T)$ and projecting out extra variables, and the optimum value of the corresponding linear program (formal definitions can be found in Section 2).

Theorem 1. *Let T be a tournament. The relaxation $\mathsf{SA}_1(T,w)$ has worst-case integrality gap equal to 7/3.*

[1] This follows from results of Speckenmeyer [17] and Khot and Regev [9]. Speckenmeyer gave a reduction from the vertex cover problem to FVST, which actually turns out to be approximation-preserving. Khot and Regev proved that, under the Unique Games Conjecture, it is NP-hard to approximate the vertex cover problem within a ratio of $2-\varepsilon$, where $\varepsilon > 0$ is any constant.

Theorem 1 allows us to replace the $\mathcal{T}_7$-relaxation used in Mnich et al.'s 7/3-approximation algorithm for FVST [14] with the Sherali–Adams relaxation $\mathsf{SA}_1(T, w)$. This significantly improves the running time of the first phase of the algorithm. Moreover, we further simplify the second phase of the algorithm, obtaining the following.

Theorem 2. *Algorithm 1 is a 7/3-approximation algorithm for* FVST. *More precisely, the algorithm outputs in polynomial time a feedback vertex set X such that* $w(X) \leq \frac{7}{3}\mathsf{SA}_1(T, w) \leq \frac{7}{3}\mathrm{OPT}(T, w)$.

We will not prove Theorem 1 directly, but derive it as a consequence of Theorem 2. Precise definitions will be given later. For now, we give a sketch of Algorithm 1, and explain how it compares with Mnich et al. [14].

Comparison to previous work

Our approach simplifies both phases of Mnich et al.'s algorithm [14]. In our first phase, instead of considering the $\mathcal{T}_7$-relaxation, we consider $\mathsf{SA}_1(T, w)$, the relaxation obtained from the basic LP after applying one round of the Sherali–Adams hierarchy. One can check that the two relaxations are incomparable.[2] However, we believe that the Sherali–Adams relaxation is preferable for its provably bounded integrality gap (see Theorem 1) and for its smaller size. By definition (see Section 2), $\mathsf{SA}_1(T, w)$ only has $O(n^4)$ constraints, while the $\mathcal{T}_7$-relaxation can have $\Omega(n^7)$ constraints.

The randomized algorithm by Lokshtanov et al. [13] has (worst-case) $O(n^{17})$ complexity. The feasible solution it returns is a 2-approximation with probability at least 1/2. Mnich et al. [14] do not give a precise analysis of the run time of their algorithm. However, it seems the solution of the linear programs is the bottleneck of their algorithm, and this is true also for ours. Therefore, by using a smaller LP, we obtain a speedup in run-time.

We now give an overview of Algorithm 1, whose input is a tournament T and a weight function w. Let x be an optimal solution to $\mathsf{SA}_1(T, w)$. For each coordinate x_v of x such that $x_v \geq 3/7$, we round up x_v to 1 and delete v from T. We continue the rounding using $\mathsf{SA}_0(T, w)$ (the basic relaxation), rounding up coordinates with value at least 1/2. At the end we obtain a *residual* tournament (which is still denoted by T for simplicity) such that all coordinates of an optimal solution of $\mathsf{SA}_0(T, w)$ are less than 1/2. The whole rounding is done exactly as in [14], except that we replace the $\mathcal{T}_7$-relaxation with $\mathsf{SA}_1(T, w)$.

Then, we proceed to the second phase, described by the algorithm Layers (see Section 4 for formal definitions). The idea follows Mnich et al. [14], but we obtain a few important simplifications thanks to our use of the Sherali–Adams relaxation during the first phase. We start from a minimum in-degree vertex z and build a breadth-first search (BFS) in-arborescence that partitions $V(T)$ in layers such that every triangle of T lies within three consecutive layers. Hence, a feedback vertex set for T can be obtained by including every other layer, and, for every layer i that is not picked, a set F_i that is a feedback vertex set for that layer (we call the set F_i a *local solution*).

Algorithm 1 FVST-main

Input: A tournament T and a weight function $w : V(T) \to \mathbb{Q}_{>0}$.
Output: A feedback vertex set of T of weight at most $\frac{7}{3}\mathrm{OPT}(T, w)$.

$x \leftarrow$ optimal solution to $\mathsf{SA}_1(T, w)$
$F \leftarrow \{v \in V(T) : x_v \geq 3/7\}$
if F is a FVS for T **then**
  return F
else
  $Z \leftarrow \varnothing$
  repeat
    add to Z all vertices of $T - F - Z$ that are not contained in any triangle
    $x \leftarrow$ optimal solution to $\mathsf{SA}_0(T - F - Z, w)$
    $F \leftarrow F \cup \{v \in V(T - F - Z) : x_v \geq 1/2\}$
  until $T - F - Z$ is empty or $x_v < 1/2$ for all $v \in V(T - F - Z)$
  $F' \leftarrow$ Layers$(T - F - Z, w, \varnothing, V(T - F - Z))$
  return $F \cup F'$
end if

The main difference with the layering algorithm of [14] is how local solutions are selected: while they use $\mathrm{CDZ}(T, w)$ on each layer as a subroutine to optimally select local solutions, we only use $\mathrm{CDZ}(T, w)$ on a small number of layers produced by the BFS procedure (see Section 4 for further details). For the other layers, we show that they can be naturally partitioned into two acyclic subtournaments. Hence, we can choose the cheapest of the two as our local solution.

Our method gives, in fact, an improved 9/4-approximation algorithm for FVST on our residual tournament, compared to the 7/3 factor obtained in Mnich et al. [14].

[2] Indeed, using for instance the code in [6], one checks that, settings unit weights ($w = \mathbf{1}$), $\mathsf{SA}_1(T, w)$ is stronger than the $\mathcal{T}_7$-relaxation when $T \in \mathcal{T}_5$, and weaker for some $T \in \mathcal{T}_7$ (specifically, for the unique "light" T_7 mentioned in Section 3).

Paper outline

In Section 2, we establish the first basic facts about the integrality gap of the basic relaxation and we define the Sherali–Adams hierarchy. In Section 3 we introduce a local structure called a *diagonal*, which will be helpful in our rounding procedure. We also classify every tournament as either *light* or *heavy*, and derive some structural properties of light tournaments. These results will be used later, since the input of our layering algorithm is a light tournament. In Section 4, we describe our layering procedure and prove several lemmas on it. Finally, in Section 5, we prove Theorems 1 and 2. A conclusion is given in Section 6.

2. The basic relaxation and the Sherali–Adams Hierarchy

We begin with a simple observation that will be useful for establishing integrality gaps of both the basic relaxation and the Sherali–Adams relaxation. Its proof is a straightforward application of the probabilistic method and can be found, for instance, in the introduction of the book by Alon and Spencer [1].

Proposition 3. *Let T be a tournament on n vertices where the direction of each arc is chosen uniformly at random. Then any feedback vertex set of T has size at least $n - O(\log n)$ with high probability.*

We now use Proposition 3 to derive the worst-case integrality gap of the basic relaxation. We give the (rather standard) proof for completeness.

Proposition 4. *The worst-case integrality gap of the basic relaxation of* FVST *is equal to 3.*

Proof. Let T be any tournament with weight function $w : V(T) \to \mathbb{Q}_{\geq 0}$, and let x^* be an optimal solution of $\max\{w^T x : x \in P(T)\}$. Let $S = \{v \in V(T) : x^*_v \geq \frac{1}{3}\}$. Then S is a feedback vertex set of weight at most $3w^T x^*$. Hence, $\mathrm{OPT}(T, w) \leq w(S) \leq 3w^T x^*$, proving that the integrality gap of T is at most 3.

On the other hand, Proposition 3 implies that there are tournaments T on n vertices with $\mathrm{OPT}(T, \mathbf{1}) = n - O(\log n)$, where we set w to be the all-ones vector. But setting $x_v = \frac{1}{3}$ for each $v \in V(T)$ yields a solution of $P(T)$ with weight $\frac{n}{3}$. Thus, taking the supremum over all such T gives that the worst case integrality gap is exactly 3. □

We now turn our attention to the Sherali–Adams relaxation. Let $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ be a polytope contained in $[0, 1]^n$ and let $P_I := \mathrm{conv}(P \cap \mathbb{Z}^n)$. Numerous optimization problems can be formulated as minimizing a linear function over P_I, where P has only a polynomial number of constraints. For example, let T be a tournament and $w : V(T) \to \mathbb{Q}_{\geq 0}$. Then $\mathrm{OPT}(T, w)$ is simply the minimum of $w^\intercal x$ over P_I, where $P = P(T)$ is the basic relaxation defined above.

The Sherali–Adams hierarchy [16] is a simple but powerful method to obtain a refining sequence of approximations for P_I. Since it does not require any knowledge of the structure of P_I, it is widely applicable. The procedure comes with a parameter r, which specifies the accuracy of the approximation. That is, for each $r \in \mathbb{N}$, we define a polytope $\mathsf{SA}_r(P)$. These polytopes satisfy $P = \mathsf{SA}_0(P) \supseteq \mathsf{SA}_1(P) \supseteq \cdots \supseteq \mathsf{SA}_r(P) \supseteq \cdots \supseteq P_I$.

An important property of the procedure is that if P is described by a polynomial number of constraints and r is a constant, then $\mathsf{SA}_r(P)$ is also described by a polynomial number of constraints (in a higher dimensional space). Therefore, for NP-hard optimization problems (such as FVST), one should not expect that $\mathsf{SA}_r(P) = P_I$ for some constant r. However, as we will see, good approximations of P_I can be extremely useful if we want to *approximately* optimize over P_I. Indeed, despite some recent results [2,8,11,15,18], we feel that the Sherali–Adams hierarchy is underutilized in the design of approximation algorithms, and hope that our work will inspire further applications.

Here is a formal description of the procedure. Let $P = \{x \in \mathbb{R}^n \mid Ax \geq b\} \subseteq [0, 1]^n$ and $r \in \mathbb{N}$. Let N_r be the *nonlinear* system obtained from P by multiplying each constraint by $\prod_{i\in I} x_i \prod_{j\in J}(1-x_j)$ for all disjoint subsets I, J of $\{1, \ldots, n\}$ such that $1 \leq |I| + |J| \leq r$. Note that if $x_i \in \{0, 1\}$, then $x_i^2 = x_i$. Therefore, we can obtain a *linear* system L_r from N_r by setting $x_i^2 := x_i$ for all $i = 1, \ldots, n$ and then $x_I := \prod_{i\in I} x_i$ for all $I \subseteq \{1, \ldots, n\}$ with $|I| \geq 2$. We then let $\mathsf{SA}_r(P)$ be the projection of L_r onto the variables x_i, $i \in \{1, \ldots, n\}$.

We let $\mathsf{SA}_r(T) := \mathsf{SA}_r(P(T))$, where $P(T)$ is the basic relaxation. In particular, we have $\mathsf{SA}_0(T) := P(T)$.

For the remainder of the paper, we only need the inequalities defining $\mathsf{SA}_1(T)$, which we now describe. Recall that $\triangle(T)$ is the collection of all $\{a, b, c\} \subseteq V(T)$ that induce a directed triangle in T. For simplicity, we call the elements of $\triangle(T)$ *triangles*. For all $abc \in \triangle(T)$ and $d \in V(T - a - b - c)$, we have the inequalities

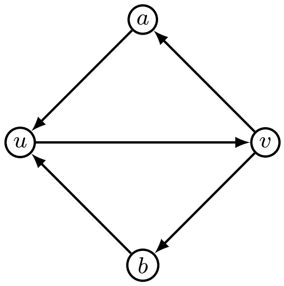
$$x_a + x_b + x_c \geq 1 + x_{ab} + x_{bc}\,, \tag{1}$$

$$x_{ad} + x_{bd} + x_{cd} \geq x_d \quad \text{and} \tag{2}$$

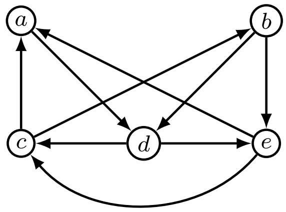
$$x_a + x_b + x_c + x_d \geq 1 + x_{ad} + x_{bd} + x_{cd}\,. \tag{3}$$

In addition, there are the inequalities

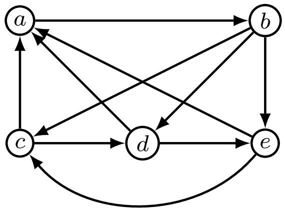
$$1 \geq x_a \geq x_{ab} \geq 0 \tag{4}$$



(A) The (unordered) pair ab is a diagonal.



(B) The two orientations of ab give different light tournaments in $\mathcal{T}_5$.



(C) The unique heavy tournament in $\mathcal{T}_5$. Note that triangle dec is heavy.

Fig. 1. An example of a diagonal and the three tournaments (up to isomorphism) in $\mathcal{T}_5$.

for all distinct $a, b \in V(T)$. Let $E(T)$ be the set of all unordered pairs of vertices of T. The polytope $\mathsf{SA}_1(T)$ is the set of all $(x_a)_{a\in V(T)} \in \mathbb{R}^{V(T)}$ such that there exists $(x_{ab})_{ab\in E(T)} \in \mathbb{R}^{E(T)}$ so that inequalities (1)–(4) are satisfied. Finally, for $w : V(T) \to \mathbb{Q}_{\geq 0}$ as above, and $r \in \{0, 1\}$, we denote by $\mathsf{SA}_r(T, w)$ the linear program $\min\{w^T x : x \in \mathsf{SA}_r(T)\}$. We will abuse notation and write $\mathsf{SA}_r(T', w)$ for a subtournament T' of T, where coordinates of w corresponding to $V(T) \setminus V(T')$ are ignored.

3. Diagonals and light tournaments

Let T be a tournament. An (unordered) pair of vertices ab is a *diagonal* if there are vertices u, v such that $uva \in \triangle(T)$ and $uvb \in \triangle(T)$ (see Fig. 1(a)). We say that a triangle *contains a diagonal* if at least one of its pairs of vertices is a diagonal, and a triangle is *heavy* if it contains at least two diagonals. A tournament T is *heavy* if at least one of its triangles is heavy. If a tournament is not heavy, we say that it is *light*.

Lemma 5. *Let T be a tournament and let $x \in \mathsf{SA}_1(T)$. If $x_v < 3/7$ for all $v \in V(T)$, then T is light.*

Proof. First, let ab be a diagonal of T. We claim that $x_{ab} \geq 1/7$. Indeed, since ab is a diagonal there must be $u, v \in V(T)$ with $uva, uvb \in \triangle(T)$. From (1), $x_a + x_u + x_v \geq 1 + x_{au} + x_{av}$ and from (2), $x_{ab} + x_{au} + x_{av} \geq x_a$. Adding these two inequalities, we obtain $x_u + x_v + x_{ab} \geq 1$, implying our claim.

Now, suppose by contradiction that T is a heavy tournament. Hence, there exists $abc \in \triangle(T)$ such that ab and bc are diagonals. By (1), we have $x_a + x_b + x_c \geq 1 + x_{ab} + x_{bc}$. By the above claim, $x_{ab} \geq 1/7$ and $x_{bc} \geq 1/7$, making the right hand side at least 9/7. So $\max(x_a, x_b, x_c) \geq 3/7$, a contradiction. □

Next we prove some results connecting light tournaments to the work of Mnich et al. [14], which relies on tournaments being $\mathcal{T}_7$-free. First, it is easy to describe the tournaments in $\mathcal{T}_5$: we refer to Figs. 1(c) and 1(b) for the proof of the following statement.

Proposition 6. *The set $\mathcal{T}_5$ contains three tournaments: one of them is heavy, while the other two are light and can be obtained from each other by reversing the orientation of one arc.*

Moreover, although we do not use this fact, we have a computer-assisted proof which shows that 120 out of 121 of the tournaments in $\mathcal{T}_7$ are heavy, and only one is light. We refer to [6] for the code of our proof. Thus, even though a light tournament is not necessarily $\mathcal{T}_7$-free, the property of being light forbids almost all of the tournaments in $\mathcal{T}_7$ as subtournaments.

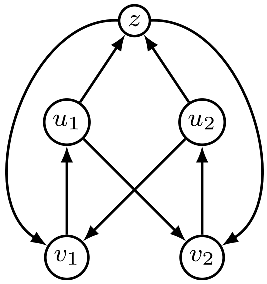


Fig. 2. $\mathcal{S}_5$ is the following subset of $\mathcal{T}_5$, where the missing arcs can be oriented arbitrarily. Two of these orientations produce isomorphic tournaments.

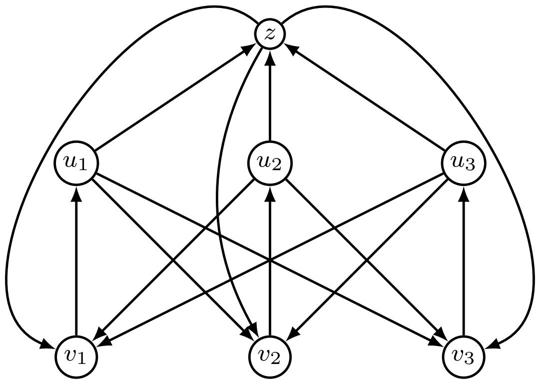


Fig. 3. $\mathcal{S}_7$ is the following subset of $\mathcal{T}_7$, where the missing arcs can be oriented arbitrarily.

We now establish further properties of light tournaments, which we will need in Section 4. Let $\mathcal{S}_5 \subseteq \mathcal{T}_5$ and $\mathcal{S}_7 \subseteq \mathcal{T}_7$ be the collection of tournaments defined in Figs. 2, and 3, respectively. If T is a tournament, we let $A(T)$ be the set of arcs of T.

Lemma 7. *Every $S \in \mathcal{S}_5$ is either heavy or has $(u_i, u_{3-i}), (v_i, v_{3-i}) \in A(S)$ for some $i \in \{1, 2\}$ (where S is labeled as in Fig. 2).*

Proof. Suppose $(u_1, u_2), (v_2, v_1) \in A(S)$. Observe that zv_2 is a diagonal, since v_1u_1z and $v_1u_1v_2$ are triangles, and v_2u_2 is a diagonal, since $v_1u_1v_2$ and $v_1u_1u_2$ are triangles. Because zv_2 and v_2u_2 are both diagonals, we conclude that the triangle v_2u_2z is heavy. The result follows by symmetry. □

Lemma 8. *Every $S \in \mathcal{S}_7$ is heavy.*

Proof. Suppose some $S \in \mathcal{S}_7$ is light, where S is labeled as in Fig. 3. By symmetry, we may assume that $(u_1, u_2), (u_2, u_3) \in A(S)$. By Lemma 7, $(v_1, v_2), (v_2, v_3) \in A(S)$. Therefore, u_2z is a diagonal, since v_1u_1z and $v_1u_1u_2$ are triangles, and zv_2 is a diagonal, since v_3u_3z and $v_3u_3v_2$ are triangles. We conclude that v_2u_2z is a heavy triangle, which contradicts that S is light. □

4. The layering procedure

This section describes our layering algorithm and proves its correctness. Lemmas 10 to 12 ensure that the algorithm actually produces a feedback vertex set. Lemmas 14 to 17 prove that Algorithm 2 is a 9/4-approximation algorithm.

We first give some notation. Let T be a light tournament with weight function $w : V(T) \to \mathbb{Q}_{\geq 0}$. For $S \subseteq V(T)$, the *in-neighborhood* of S is $N(S) := \{v \notin S \mid (v, u) \in A(T) \text{ for some } u \in S\}$ and $N(u) := N(\{u\})$. For every $z \in V(T)$, define $V_1(z) = \{z\}$, and for $i \geq 2$ let $V_{i+1}(z) := N(\bigcup_{j=1}^{i} V_j(z))$. In other words, $V_i(z)$ is the set of vertices whose shortest directed path to z has length exactly $i - 1$.

Given two sets $S, Z \subseteq V(T)$, we say that Z *in-dominates* S if for every $s \in S$ there is a $z \in Z$ with $(s, z) \in A(T)$. 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 (we call Z' a *2-in-dominating set*).

We now give a more detailed description of the LAYERS algorithm introduced in Section 1, referring to Algorithm 2 for the pseudo-code. First, our input tournament T is required to be light, which can be assumed thanks to Lemma 5. This will be crucial for the correctness of the algorithm.

We begin by picking a vertex z of minimum in-degree, and by defining layers $V_1 := V_1(z) = \{z\}$, $V_2 := V_2(z), \ldots, V_k := V_k(z)$, until we have included all vertices from which z can be reached. If these layers do not cover all the vertices of T, we just re-run the procedure on the remaining vertices by choosing a new vertex z (we refer to this as a *fresh start*). We remark that any triangle in T lies within three consecutive layers. Hence, a feedback vertex set for T can be obtained by including every other layer and, for every layer V_i that is not picked, a local solution F_i that is a feedback vertex set for V_i.

We are now left to explain how we obtain the local solutions F_i. The first layer V_1 is a singleton, hence, setting $F_1 = \emptyset$ suffices. The same applies to other "first" layers after a fresh start. For the second layer V_2, and for the other "second" layers after a fresh start, we will get a local solution via the algorithm CDZ from Cai et al. [5], following the same idea as the layering algorithm of Mnich et al. [14]. This is justified by the fact that the subtournament corresponding to those layers is $\mathcal{T}_5$-free (See Lemma 11).

For all the other layers, a different property is established, which allows us to immediately obtain our local solutions. Indeed, we will show (Lemmas 9, 12) that a layer $V_i(z)$ with $i \geq 3$ is 2-in-dominated by the previous layer $V_{i-1}(z)$. We remark that such a 2-in-dominating set $\{z_{i-1}, z'_{i-1}\}$ can be found efficiently, for instance by trying all possible pairs: this procedure is called 2-IN-DOMINATES in Algorithm 2. Then, it turns out (Lemma 10) that $V_i(z)$ can be partitioned into two subtournaments, $U_i = N(z_{i-1})$ and $S_i = V_i(z) - U$, that are both acyclic. This implies that our local solution F_i can be set as the cheapest of U_i and S_i.

Algorithm 2 LAYERS(T, w, U_i, W)

Input: A light tournament T, $w : V(T) \to \mathbb{Q}_{\geq 0}$, the current root layer U_i , and the set of vertices W that have not been processed yet ($U_0 := \varnothing$ and $W := V(T)$ on the first call). We assume all objects that depend on i (including i itself) to be available throughout subsequent recursive calls.
Output: A feedback vertex set F' of T of weight at most $\frac{3}{4}w(T)$.

if $W = \varnothing$ **then** {Finished}
  $L_0 \leftarrow \cup_{j \text{ even}} U_j \cup S_j$, $L_1 \leftarrow \cup_{j \text{ odd}} U_j \cup S_j$
  $F' \leftarrow (\cup_{j=1}^{i} F_{2j}) \cup L_1$ if $w(L_0) \geq w(L_1)$ otherwise $(\cup_{j=0}^{i-1} F_{2j+1}) \cup L_0$
  return F'
end if
if $N(U_i) \neq \varnothing$ **then**
  $\{z_i, z'_i\} \leftarrow$2-IN-DOMINATES$(N(U_i))$ with $w(N(z_i) \cap W) \geq w(N(z'_i) \cap W)$
  $U_{i+1} \leftarrow N(z_i) \cap W$, $S_{i+1} \leftarrow N(z'_i) \cap W - U_{i+1}$, $W \leftarrow W - U_{i+1} - S_{i+1}$
  $F_{i+1} = S_{i+1}$
  $i \leftarrow i + 1$
  return LAYERS(T, w, U_{i+1}, W)
else {Fresh Start}
  $z_{i+1} \leftarrow$ choose $z \in W$ with $|N(z) \cap W|$ minimum
  $U_{i+1} \leftarrow \{z_{i+1}\}$, $U_{i+2} \leftarrow N(z_{i+1}) \cap W$, $S_{i+1} \leftarrow \varnothing$
  $F_{i+1} \leftarrow \varnothing$
  $F_{i+2} \leftarrow$ CDZ(U_{i+2}, w)
  $W \leftarrow W - (U_{i+1} \cup U_{i+2})$
  $i \leftarrow i + 2$
  return LAYERS(T, w, U_{i+2}, W)
end if

We now prove the lemmas needed to show the correctness of our algorithm. For simplicity, we will denote all layers produced by Algorithm 2 as $V_i = U_i \cup S_i$, for $i = 1, \ldots, \ell$, where S_i is possibly empty (for instance, in layers corresponding to a fresh start).

Lemma 9. *For an arbitrary vertex z in a light tournament T, $V_3(z)$ is 2-in-dominated by $V_2(z)$.*

Proof. Let $H = \{h_1, h_2, \ldots, h_k\} \subseteq V_2(z)$ be an inclusion-wise minimal set that in-dominates $V_3(z)$. Suppose $k \geq 3$. By minimality, for each $h_i \in H$ there must be some $v_i \in V_3(z)$ such that $(v_i, h_i) \in A(T)$ and $(h_i, v_j) \in A(T)$ for all $j \neq i$. Since $(z, v_i) \in A(T)$ for all i, it follows that $T[\{z, h_1, h_2, h_3, v_1, v_2, v_3\}]$ is isomorphic to a tournament in $\mathcal{S}_7$ (see Fig. 3). Therefore, by Lemma 8, $T[\{z, h_1, h_2, h_3, v_1, v_2, v_3\}]$ is heavy, which contradicts that T is light. □

Lemma 10. *Let T be a light tournament, z be any vertex of T, and $i \geq 3$. If $V_i(z)$ is 2-in-dominated by $\{z_{i-1}, z'_{i-1}\} \subseteq V_{i-1}(z)$ (possibly $z_{i-1} = z'_{i-1}$), then $U := N(z_{i-1}) \cap V_i(z)$ and $S := V_i(z) - U$ are triangle-free.*

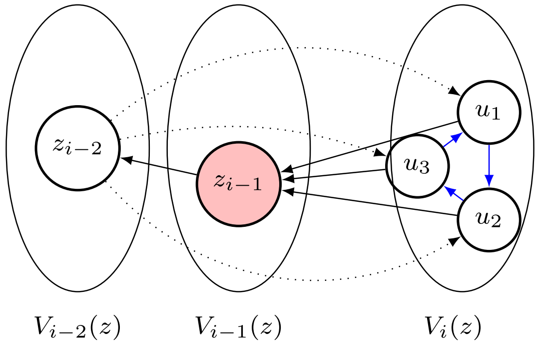


Fig. 4. The triangle $u_1u_2u_3$ cannot exist in a light tournament. All of its arcs are diagonals due to triangles $vz_{i-1}z_{i-2}$ for $v \in \{u_1, u_2, u_3\}$.

Proof. Suppose by contradiction that $u_1u_2u_3$ is a triangle in U (see Fig. 4). Since $z_{i-1} \in V_{i-1}(z)$ and $i \geq 3$, we have $(z_{i-1}, r) \in A(T)$ for some $r \in V_{i-2}(z)$. Since $U \subseteq V_i(z)$, arcs $(r, u_1), (r, u_2), (r, u_3) \in A(T)$. Thus, ru_jz_{i-1} is a triangle for all $j = 1, 2, 3$. It follows that the triangle $u_1u_2u_3$ is heavy, since all of its arcs are diagonals, a contradiction. If S has a triangle, we can repeat the same argument. □

The next lemma ensures that the layer produced after a fresh start is $\mathcal{T}_5$-free. This allows us to use the algorithm CDZ from Cai et al. [5], as described above. Its proof follows the proof of Lemma 9 of Mnich et al. [14], except that we assume that T is light.

Lemma 11. *Let z be a minimum in-degree vertex in a light tournament T. Then $V_2(z)$ is $\mathcal{T}_5$-free.*

Proof. We assume $V_2(z) \neq \varnothing$, otherwise there is nothing to show, and we suppose by contradiction that $X \subseteq V_2(z)$ is a light $\mathcal{T}_5$ (X cannot be heavy as T is light, so X is oriented as in Fig. 1(b)). For every $u \in V_2(z)$ there must be a $v \in V_3(z)$ with $(v, u) \in A(T)$. If not then $N(u) \subsetneq V_2(z) = N(z)$, contradicting the minimality of $|N(z)|$. Thus $V_3(z) \neq \varnothing$. Let $H \subseteq V_3(z)$ be an inclusion-wise minimal subset of $V_3(z)$ such that for every $u \in V_2(z)$ there exists $v \in H$ with $(v, u) \in A(T)$. We distinguish cases according to the size of H.

Case 1: $H = \{h\}$. Then hu_iz are triangles for all $u_i \in X$, therefore all arcs in X are diagonals. Since X must contain at least some triangle, this triangle must be heavy, since all of its arcs are diagonals, contradicting the fact that T is light.

Case 2: $H = \{f, h\}$. Let $X = \{a, b, c, d, e\}$. We can assume without loss of generality that f points to exactly three vertices of X, for the following reason. If there are less than three, we can swap h with f. If there are more than three, then f must point to a triangle of X (since $T[X]$ is a $\mathcal{T}_5$-subtournament), which would be heavy, arguing as in Case 1.

Notice that ec and de are diagonals within X (due to triangles ade and adc, bdc and bec, respectively), hence, none of ad, ea, cb, be can be diagonals, otherwise one of ade or cbe will be a heavy triangle. This implies that f cannot point to both vertices of any of the latter pairs. From this, one easily derives that f cannot point to a nor b. Hence, (a, f), (b, f), (f, d), (f, e), $(f, c) \in A(T)$, which implies $(h, a), (h, b) \in A(T)$. This forces $(e, h), (c, h), (d, h) \in A(T)$; otherwise, again, one of ad, ea, cb, be is a diagonal. See Fig. 5 for the orientations we have determined thus far. Notice that adc and fca are triangles, so fd is a diagonal. Moreover, since had and zha are triangles, dz is a diagonal. Therefore zfd is a heavy triangle, a contradiction.

Case 3: $|H| \geq 3$. In this case, one can easily find a tournament in $\mathcal{S}_7$ made of z, three vertices of $V_2(z)$ and three vertices of $V_3(z)$, in contradiction with Lemma 8 (see the proof of Lemma 9). □

The next lemma ensures that, at each recursive call of Algorithm 2, we can find local solutions by either applying CDZ(T, w) or Lemma 10. Together with the previous lemmas, it is enough to conclude that Algorithm 2 outputs a feedback vertex set of our (light) tournament T. For simplicity, we refer to layer V_1 and to other layers corresponding to a fresh start as *fresh start layers*.

Lemma 12. *Let $V_1 = U_1 \cup S_1, \ldots, V_\ell = U_\ell \cup S_\ell$ be the layers produced by Algorithm 2, run on input $(T, w, U_0 := \varnothing, W := V(T))$. For $i = 1, \ldots, \ell - 1$, if V_i is a fresh start layer, then $T[V_{i+1}] = T[U_{i+1}]$ is $\mathcal{T}_5$-free; otherwise, then S_{i+1} and U_{i+1} are both feedback vertex sets of $T[V_{i+1}]$.*

Proof. Let $i \in \{1, \ldots, \ell - 1\}$. If V_i is a fresh start layer, U_{i+1} is equal to $V_2(z)$ for some $z \in V(T)$ (see line 14 of Algorithm 2). Therefore, by Lemma 11, $T[U_{i+1}] = T[V_2(z)]$ is $\mathcal{T}_5$-free.

If V_i is not a fresh start layer, then there is some vertex $z_{i-1} \in V(T)$ such that $U_{i+1} \cup S_{i+1} \subseteq V_3(z_{i-1})$. By Lemma 9, $N(U_i)$ 2-in-dominates $U_{i+1} \cup S_{i+1}$. Therefore, by Lemma 10, S_{i+1} and U_{i+1} are both triangle-free. Thus, S_{i+1} and U_{i+1} are both feedback vertex sets of $T[V_{i+1}]$. □

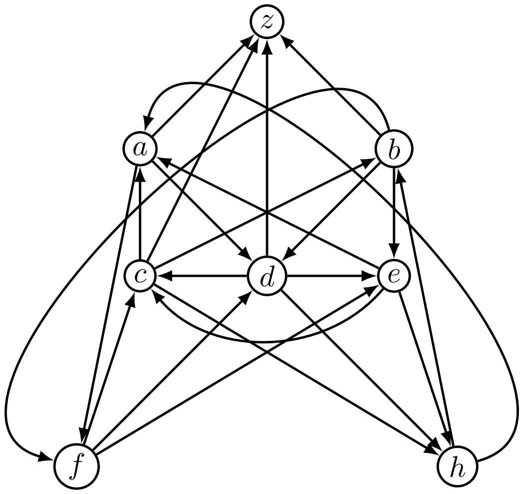


Fig. 5. The orientations determined by the proof of Lemma 11.

After having shown the correctness of Algorithm 2, we focus on bounding the approximation ratio of its output. This mostly amounts to bounding the weight of the local solutions obtained during the algorithm. We first summarize in a proposition the results from Cai et al. [5] that we need.

Proposition 13 (*[5]*)**.** *Let T be a $\mathcal{T}_5$-free tournament. Then the basic relaxation $P(T)$ is integral, and the set $F = \mathrm{CDZ}(T, w)$ obtained by solving the linear program $\mathsf{SA}_0(T, w)$ is a minimum weight feedback vertex set.*

Lemma 14. *Let $F_1, \ldots, F_\ell$ and $U_0, \ldots, U_\ell$ be the sets produced by Algorithm 2, run on input $(T, w, U_0 := \varnothing, W := V(T))$. Then for $i = 1, \ldots, \ell$, $w(F_i) \leq w(N(U_{i-1}))/2$.*

Proof. If $F_i = \emptyset$ (for instance if $i = 1$), then the lemma clearly holds. If layer V_{i-1} is not a fresh start layer, then we have $w(F_i) \leq w(N(U_{i-1}))/2$ by construction (see line 9 of Algorithm 2).

Thus, we may suppose that layer V_{i-1} is a fresh start layer. Hence, $V_{i-1} = U_{i-1} = \{z_{i-1}\}$, $U_i = N(z_{i-1})$, and $F_i = \mathrm{CDZ}(U_i, w)$ (see line 16 of Algorithm 2). By Lemma 12, $T[U_i]$ is $\mathcal{T}_5$-free, and by Proposition 13, F_i is a minimum weight feedback vertex set of $T[U_i]$. Since the all $\frac{1}{3}$-vector is feasible for the basic relaxation of $T[U_i]$, and this relaxation is integral by Proposition 13,

$$w(F_i) \leq \frac{1}{3}w(U_i) = \frac{1}{3}w(N(U_{i-1})) \leq \frac{1}{2}w(N(U_{i-1})),$$

as required. □

In the next two lemmas, we assume that Algorithm 1 is run on input (T, w), and we establish properties of the sets defined within the algorithm during the first phase (lines 1-11 of Algorithm 1). We briefly recall the first phase: it starts by optimally solving $SA_1(T, w)$, rounds up to 1 the coordinates larger than 3/7, then iteratively solves the basic relaxation $SA_0(T, w)$ on the residual tournament rounding up coordinates larger than 1/2, until all coordinates in an optimal solution are strictly less than 1/2. Whenever we round up a coordinate, we include the corresponding vertex in set F. Moreover, whenever a vertex does not belong to any directed triangle of the current tournament, we include it in set Z (those vertices do not belong to any minimal FVS and can be ignored).

Lemma 15. *After the first phase of Algorithm 1,*

$$w(F) \leq \frac{7}{3}\big(\mathsf{SA}_1(T, w) - \mathsf{SA}_0(T - F - Z, w)\big).$$

Proof. We first consider the set F obtained after the first rounding step (i.e. before line 7 of Algorithm 1). Letting x denote the optimal solution to $\mathsf{SA}_1(T, w)$, we get

$$w(F) \leq \frac{7}{3}\sum_{v \in F} w(v)x_v$$

$$= \frac{7}{3}\left(\sum_{v\in V(T)} w(v)x_v - \sum_{v\in V(T-F)} w(v)x_v\right)$$
$$\leq \frac{7}{3}\big(\mathsf{SA}_1(T,w) - \mathsf{SA}_1(T-F,w)\big)$$
$$\leq \frac{7}{3}\big(\mathsf{SA}_1(T,w) - \mathsf{SA}_0(T-F,w)\big)\,,$$

where the second inequality follows from the fact that x, restricted to $V(T-F)$, is a feasible solution of $\mathsf{SA}_1(T-F,w)$, and the third inequality follows, since the first round of Sherali–Adams is at least as strong as the zeroth.

Now, we show that the above inequality is preserved after successive rounding steps. Consider any such step, and let F, Z be the sets obtained before that step, and let F', Z' be obtained after the step (see lines 7–11 of Algorithm 1). Recall that $F \subseteq F', Z \subseteq Z'$. In order to conclude the proof, we show that if

$$w(F) \leq \frac{7}{3}\big(\mathsf{SA}_1(T,w) - \mathsf{SA}_0(T-F-Z,w)\big),$$

then

$$w(F') \leq \frac{7}{3}\big(\mathsf{SA}_1(T,w) - \mathsf{SA}_0(T-F'-Z',w)\big).$$

Let x denote the optimal solution to $\mathsf{SA}_0(T-F-Z,w)$. For simplicity, we restrict to the case where F' is obtained from F by adding a single vertex v (such that $x_v \geq \frac{1}{2}$): the general case follows by the same argument.

We have that x, restricted to coordinates $V(T-F'-Z')$, is a feasible solution of $\mathsf{SA}_0(T-F'-Z',w)$. Hence,

$$\mathsf{SA}_0(T-F-Z,w) \geq w(v)x_v + \mathsf{SA}_0(T-F'-Z',w),$$

since the vertices in Z' do not affect the optimal value $\mathsf{SA}_0(T-F'-Z',w)$. But then we have:

$$w(v) \leq 2w(v)x_v \leq 2\big(\mathsf{SA}_0(T-F-Z,w) - \mathsf{SA}_0(T-F'-Z',w)\big) \leq \frac{7}{3}\big(\mathsf{SA}_0(T-F-Z,w) - \mathsf{SA}_0(T-F'-Z',w)\big),$$

which is easily seen to imply our claim since $F' = F \cup \{v\}$. □

Lemma 16. *After the first phase of Algorithm 1,*

$$\mathsf{SA}_0(T-F-Z,w) = w(T-F-Z)/3.$$

Proof. The proof is the same as [14, Lemma 6], and due to [10], but we include it here for completeness. Let $T' = T-F-Z$. Suppose $x_v = 0$ for some $v \in V(T')$. Since every vertex of T' is contained in a triangle, v is in some triangle vab of T. Thus, $x_a + x_b \geq 1$, and so $\max(x_a, x_b) \geq 1/2$, which contradicts that neither a nor b are in F. Thus $x_v > 0$ for all $v \in V(T')$. Now, notice that the following is the dual linear program of $\mathsf{SA}_0(T',w)$: $\max\{\mathbf{1}^T y : \sum_{\triangle:u\in\triangle} y_\triangle \leq w_u \text{ for } u \in V(T'), y \geq 0\}$, where y has a coordinate for each triangle of T'. By primal–dual slackness, if y is an optimal solution to the dual, then $\sum_{\triangle:u\in\triangle} y_\triangle = w_u$ for all $u \in V(T')$. Therefore,

$$w(V(T')) = \sum_{u\in V(T')}\sum_{\triangle:u\in\triangle} y_\triangle = \sum_{\triangle\in\Delta(T')} y_\triangle \sum_{u\in\triangle} 1 = 3\sum_{\triangle\in\Delta(T')} y_\triangle = 3\mathsf{SA}_0(T',w),$$

as required. □

Lemma 17. *Let F' be the set output by Algorithm 2 on input $(T' := T-F-Z, w, U_0 := \varnothing, W = V(T'))$. Then $F \cup F'$ is a feedback vertex set of T and $w(F') \leq \frac{9}{4}\mathsf{SA}_0(T',w)$.*

Proof. Algorithm 2 partitions $V(T')$ into layers $S_i \cup U_i$ for $i = 1,\dots,\ell$, for some $\ell > 0$. By symmetry, we may assume that the total weight of the even layers is at least the total weight of the odd layers. That is, $w(L_0) \geq w(L_1)$, using the notation of the algorithm. Then the output F' consists of the union of all odd layers, i.e. L_1, and of the sets F_i, for i even. By construction, F_i is an FVS of $T'[S_i \cup U_i]$, for each i. Since all triangles in T' are contained in three consecutive layers, F' is an FVS of T'. Hence, $F \cup F'$ is an FVS of T. Moreover, since $w(F_i) \leq w(S_i \cup U_i)/2$ for each $i = 1,\dots,\ell$, we have $w(\cup_{j \text{ even}} F_j) \leq \frac{1}{2}w(L_0)$

$$w(F') = w(L_1) + w(\cup_{j \text{ even}} F_j) \leq w(V(T')) - w(L_0) + \frac{1}{2}w(L_0) \leq \frac{3}{4}w(V(T')) = \frac{9}{4}\mathsf{SA}_0(T',w),$$

where the last equality follows from Lemma 16. □

5. Proofs of the main theorems

Given the results we have already established, it is now easy to prove the correctness of Algorithm 1.

Proof of Theorem 2. By Lemma 17, $F \cup F'$ is a feedback vertex set of T. It remains to show the approximation guarantee. Recall that $F = \{v : x_v \geq 3/7\}$ where x is an optimal solution for $\mathsf{SA}_1(T, w)$. By Lemma 15, $w(F) \leq \frac{7}{3}(\mathsf{SA}_1(T, w) - \mathsf{SA}_0(T - F - Z, w))$. Since x restricted to $T - F - Z$ is feasible for $\mathsf{SA}_1(T - F - Z)$, Lemma 17 implies that $w(F') \leq \frac{9}{4}\mathsf{SA}_0(T - F - Z, w) \leq \frac{7}{3}\mathsf{SA}_0(T - F - Z, w)$. Adding these two inequalities yields

$$w(F') + w(F) \leq \frac{7}{3}\mathsf{SA}_1(T, w) \leq \frac{7}{3}\mathrm{OPT}(T, w),$$

as required. □

We are now ready to derive Theorem 1 from Theorem 2.

Proof of Theorem 1. The fact that the integrality gap of SA_1 for FVST is at most 7/3 follows from Theorem 2. For the other inequality, on one hand note that for every tournament T, the all $\frac{3}{7}$-vector is feasible for $\mathsf{SA}_1(T)$ (by setting $x_{uv} = \frac{1}{7}$ for all uv). On the other hand, Proposition 3 shows that for a random n-node tournament T, $\mathrm{OPT}(T, \mathbf{1}_T) = n - O(\log n)$ with high probability. □

6. Conclusion

In this paper we give a simple 7/3-approximation algorithm for FVST, based on performing just one round of the Sherali–Adams hierarchy on the basic relaxation. It is a bit of a miracle that $\mathsf{SA}_1(T)$ already "knows" a remarkable amount of structure about feedback vertex sets in tournaments. It is unclear how much more knowledge $\mathsf{SA}_r(T)$ acquires as r increases, but our approach naturally begs the question of whether performing a constant number of rounds of Sherali–Adams leads to a 2-approximation for FVST. This would solve the main open question from Lokshtanov et al. [13].

We suspect that performing more rounds does improve the approximation ratio, but the analysis becomes more complicated. Indeed, it could be that $\mathsf{SA}_2(T)$ already gives a 9/4-approximation algorithm for FVST, since our layering procedure has a 9/4-approximation factor. Note that $\mathsf{SA}_2(T)$ does contain new inequalities such as $x_a + x_b + x_c \geq 1 + x_{ab} + x_{ac} + x_{cb} - x_{abc}$, for all $abc \in \triangle(T)$, which may be exploited.

As further evidence, for the related problem of *cluster vertex deletion* [2], we showed that one round of Sherali–Adams has an integrality gap of 5/2, and for every $\varepsilon > 0$ there exists $r \in \mathbb{N}$ such that r rounds of Sherali–Adams has integrality gap at most $2+\varepsilon$. Indeed, this work can be seen as unifying the approaches of Mnich et al. [14] and some of the polyhedral results of our work on cluster vertex deletion [2,3].

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors would like to sincerely thank the reviewers for their useful comments, that improved the presentation of the paper.

References

[1] N. Alon, J.H. Spencer, The Probabilistic Method, fourth ed., Wiley Publishing, 2016, pp. 38–39.
[2] M. Aprile, M. Drescher, S. Fiorini, T. Huynh, A tight approximation algorithm for the cluster vertex deletion problem, in: Integer Programming and Combinatorial Optimization: 22nd International Conference, IPCO 2021, Atlanta, GA, USA, May 19–21, 2021, Proceedings 22, Springer, 2021, pp. 340–353.
[3] M. Aprile, M. Drescher, S. Fiorini, T. Huynh, A tight approximation algorithm for the cluster vertex deletion problem, Math. Program. (2022) 1–23.
[4] J.S. Banks, Sophisticated voting outcomes and agenda control, Soc. Choice Welf. 1 (4) (1985) 295–306.
[5] M.-c. Cai, X. Deng, W. Zang, An approximation algorithm for feedback vertex sets in tournaments, SIAM J. Comput. 30 (6) (2001) 1993–2007.
[6] M. Drescher, FVST, 2022, GitHub, GitHub repository: https://github.com/knavely/FVST.
[7] A. Freund, R. Bar-Yehuda, K. Bendel, Local ratio: a unified framework for approximation algorithms, ACM Comput. Surv. 36 (2005) 422–463.
[8] S. Garg, Quasi-PTAS for scheduling with precedences using LP hierarchies, in: 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2018, p. 9063.
[9] S. Khot, O. Regev, Vertex cover might be hard to approximate to within 2- ε, J. Comput. System Sci. 74 (3) (2008) 335–349.
[10] M. Krivelevich, On a conjecture of Tuza about packing and covering of triangles, Discrete Math. 142 (1–3) (1995) 281–286.
[11] E. Levey, T. Rothvoss, A $(1+\epsilon)$-approximation for makespan scheduling with precedence constraints using LP hierarchies, SIAM J. Comput. 50 (3) (2019) 201–217.
[12] D. Lokshtanov, P. Misra, J. Mukherjee, F. Panolan, G. Philip, S. Saurabh, 2-Approximating feedback vertex set in tournaments, in: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2020, pp. 1010–1018.

[13] D. Lokshtanov, P. Misra, J. Mukherjee, F. Panolan, G. Philip, S. Saurabh, 2-approximating feedback vertex set in tournaments, ACM Trans. Algorithms 17 (2) (2021) 1–14.
[14] M. Mnich, V. Vassilevska Williams, L.A. Végh, A 7/3-Approximation for feedback vertex sets in tournaments, in: 24th Annual European Symposium on Algorithms, ESA 2016, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
[15] R. O'Donnell, T. Schramm, Sherali–Adams strikes back, Theory Comput. 17 (1) (2021) 1–30.
[16] H.D. Sherali, W.P. Adams, A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems, SIAM J. Discrete Math. 3 (3) (1990) 411–430.
[17] E. Speckenmeyer, On feedback problems in digraphs, in: Graph-Theoretic Concepts in Computer Science: 15th International Workshop WG'89 Castle Rolduc, the Netherlands, June 14–16, 1989 Proceedings 15, Springer, 1990, pp. 218–231.
[18] Y. Yoshida, Y. Zhou, Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems, in: ITCS'14—PRoceedings of the 2014 Conference on Innovations in Theoretical Computer Science, ACM, New York, 2014, pp. 423–437.