



## Kernels for feedback arc set in tournaments

Stéphane Bessy<sup>c</sup>, Fedor V. Fomin<sup>a</sup>, Serge Gaspers<sup>b</sup>, Christophe Paul<sup>c</sup>, Anthony Perez<sup>c,\*</sup>, Saket Saurabh<sup>d</sup>, Stéphan Thomassé<sup>c</sup>

<sup>a</sup> Department of Informatics, University of Bergen, N-5020 Bergen, Norway

<sup>b</sup> CMM, Universidad de Chile, Av. Blanco Encalada 2120, 8370459 Santiago de Chile, Chile

<sup>c</sup> LIRMM – Université Montpellier 2, CNRS, 161 rue Ada, 34932 Montpellier, France

<sup>d</sup> The Institute of Mathematical Sciences, Chennai 600 113, India

### ARTICLE INFO

#### Article history:

Received 10 January 2010

Received in revised form 9 September 2010

Accepted 4 October 2010

Available online 8 October 2010

#### Keywords:

Feedback arc set

Tournaments

Kernelization

Parameterized algorithms

Graph algorithms

### ABSTRACT

A tournament  $T = (V, A)$  is a directed graph in which there is exactly one arc between every pair of distinct vertices. Given a digraph on  $n$  vertices and an integer parameter  $k$ , the FEEDBACK ARC SET problem asks whether the given digraph has a set of  $k$  arcs whose removal results in an acyclic digraph. The FEEDBACK ARC SET problem restricted to tournaments is known as the  $k$ -FEEDBACK ARC SET IN TOURNAMENTS ( $k$ -FAST) problem. In this paper we obtain a linear vertex kernel for  $k$ -FAST. That is, we give a polynomial time algorithm which given an input instance  $T$  to  $k$ -FAST obtains an equivalent instance  $T'$  on  $O(k)$  vertices. In fact, given any fixed  $\epsilon > 0$ , the kernelized instance has at most  $(2 + \epsilon)k$  vertices. Our result improves the previous known bound of  $O(k^2)$  on the kernel size for  $k$ -FAST. Our kernelization algorithm solves the problem on a subclass of tournaments in polynomial time and uses a known polynomial time approximation scheme for  $k$ -FAST.

© 2010 Elsevier Inc. All rights reserved.

## 1. Introduction

Given a directed graph  $G = (V, A)$  on  $n$  vertices and an integer parameter  $k$ , the FEEDBACK ARC SET problem asks whether the given digraph has a set of  $k$  arcs whose removal results in an acyclic directed graph. In this paper, we consider this problem in a special class of directed graphs, *tournaments*. A tournament  $T = (V, A)$  is a directed graph in which there is exactly one directed arc between every pair of vertices. More formally the problem we consider is defined as follows.

**$k$ -FEEDBACK ARC SET IN TOURNAMENTS ( $k$ -FAST):** Given a tournament  $T = (V, A)$  and a positive integer  $k$ , does there exist a subset  $F \subseteq A$  of at most  $k$  arcs whose removal makes  $T$  acyclic.

In the weighted version of  $k$ -FAST, we are also given integer weights (each weight is at least one) on the arcs and the objective is to find a feedback arc set of weight at most  $k$ . This problem is called  $k$ -WEIGHTED FEEDBACK ARC SET IN TOURNAMENTS ( $k$ -WFAST).

Feedback arc sets in tournaments are well studied from the combinatorial [20,22,29,30,33,37], statistical [31] and algorithmic [2,3,14,26,35,36] points of view. The problems  $k$ -FAST and  $k$ -WFAST have several applications. In *rank aggregation* we are given several rankings of a set of objects, and we wish to produce a single ranking that on average is as consistent as possible with the given ones, according to some chosen measure of consistency. This problem has been studied in the context of voting [8,11,13], machine learning [12], and search engine ranking [18,19]. A natural consistency measure for rank

\* Corresponding author.

E-mail address: [perez@lirmm.fr](mailto:perez@lirmm.fr) (A. Perez).

aggregation is the number of pairs that occur in a different order in the two rankings. This leads to *Kemeny rank aggregation* [24,25], a special case of  $k$ -WFAST.

The  $k$ -FAST problem is known to be NP-complete by recent results of Alon [3] and Charbit et al. [10] while  $k$ -WFAST is known to be NP-complete by Bartholdi III et al. [5]. From an approximation perspective,  $k$ -WFAST is APX-hard [32] but admits a polynomial time approximation scheme when the edge weights are bounded by a constant [26]. The problem is also well studied in parameterized complexity. In this area, a problem with input size  $n$  and a parameter  $k$  is said to be fixed parameter tractable (FPT) if there exists an algorithm to solve this problem in time  $f(k) \cdot n^{O(1)}$ , where  $f$  is an arbitrary function of  $k$ . Raman and Saurabh [28] showed that  $k$ -FAST and  $k$ -WFAST are FPT by obtaining an algorithm running in time  $O(2.415^k \cdot k^{4.752} + n^{O(1)})$ . Recently, Alon et al. [4] have improved this result by giving an algorithm for  $k$ -WFAST running in time  $O(2^{O(\sqrt{k} \log^2 k)} + n^{O(1)})$ . This algorithm runs in sub-exponential time, a trait uncommon to parameterized algorithms. Moreover, a new algorithm due to Karpinsky and Schudy [23] with running time  $O(2^{O(\sqrt{k})} + n^{O(1)})$  improves again the complexity of  $k$ -WFAST. Finally, Fomin et al. [21] provided a sub-exponential local search algorithm for  $k$ -WFAST. In this paper we investigate  $k$ -FAST from the view point of kernelization, currently one of the most active subfields of parameterized algorithms.

A parameterized problem is said to admit a *polynomial kernel* if there is a polynomial (in  $n$ ) time algorithm, called a *kernelization* algorithm, that reduces the input instance to an instance whose size is bounded by a polynomial  $p(k)$  in  $k$ , while preserving the answer. This reduced instance is called a  $p(k)$  *kernel* for the problem. When  $p(k)$  is a linear function of  $k$  then the corresponding kernel is a linear kernel. Kernelization has been at the forefront of research in parameterized complexity in the last couple of years, leading to various new polynomial kernels as well as tools to show that several problems do not have a polynomial kernel under some complexity-theoretic assumptions [6,7,9,15,17,34]. In this paper we continue the current theme of research on kernelization and obtain a *linear vertex kernel* for  $k$ -FAST. That is, we give a polynomial time algorithm which given an input instance  $T$  to  $k$ -FAST obtains an equivalent instance  $T'$  on  $O(k)$  vertices. More precisely, given any fixed  $\epsilon > 0$ , we find a kernel with at most  $(2 + \epsilon)k$  vertices in polynomial time. The reason we call it a linear *vertex kernel* is that, even though the number of vertices in the reduced instance is at most  $O(k)$ , the number of arcs is still  $O(k^2)$ . Our result improves the previous known bound of  $O(k^2)$  on the vertex kernel size for  $k$ -FAST [4,16]. For our kernelization algorithm we find a subclass of tournaments where one can find a minimum sized feedback arc set in polynomial time (see Lemma 3.8) and use the known polynomial time approximation scheme for  $k$ -FAST by Kenyon-Mathieu and Schudy [26]. The polynomial time algorithm for a subclass of tournaments could be of independent interest.

The paper is organized as follows. In Section 2, we give some definition and preliminary results regarding feedback arc sets. In Section 3 we give a linear vertex kernel for  $k$ -FAST. Finally we conclude with some remarks in Section 4.

## 2. Preliminaries

Let  $T = (V, A)$  be a tournament on  $n$  vertices. We use  $T_\sigma = (V_\sigma, A)$  to denote a tournament whose vertices are ordered under a fixed ordering  $\sigma = v_1, \dots, v_n$  (we also use  $D_\sigma$  for an ordered directed graph). We say that an arc  $v_i v_j$  of  $T_\sigma$  is a *backward arc* if  $i > j$ , otherwise we call it a *forward arc*. Moreover, given any partition  $\mathcal{P} := \{V_1, \dots, V_l\}$  of  $V_\sigma$ , where every  $V_i$  is an interval according to the ordering of  $T_\sigma$ , we use  $A_B$  to denote all arcs between the intervals (having their endpoints in different intervals), and  $A_I$  for all arcs within the intervals. If  $T_\sigma$  contains no backward arc, then we say that it is *transitive*.

For a vertex  $v \in V$  we denote its *in-neighborhood* by  $N^-(v) := \{u \in V \mid uv \in A\}$  and its *out-neighborhood* by  $N^+(v) := \{u \in V \mid vu \in A\}$ . A set of vertices  $M \subseteq V$  is a *module* if and only if  $N^+(u) \setminus M = N^+(v) \setminus M$  for every  $u, v \in M$ . For a subset of arcs  $A' \subseteq A$ , we define  $T[A']$  to be the digraph  $(V', A')$  where  $V'$  is the union of endpoints of the arcs in  $A'$ . Given an ordered digraph  $D_\sigma$  and an arc  $e = v_i v_j$ ,  $S(e) = \{v_i, \dots, v_j\}$  denotes the *span* of  $e$ . The number of vertices in  $S(e)$  is called the *length* of  $e$  and is denoted by  $l(e)$ . Thus, for every arc  $e = v_i v_j$ ,  $l(e) = |i - j| + 1$ . Finally, for every vertex  $v$  in the span of  $e$ , we say that  $e$  is *above*  $v$ .

In this paper, we will use the well-known fact that every acyclic tournament admits a transitive ordering. In particular, we will consider *maximal transitive modules*. We also need the following result for our kernelization algorithm.

**Lemma 2.1.** (See [28].) *Let  $D = (V, A)$  be a directed graph and  $F$  be a minimal feedback arc set of  $D$ . Let  $D'$  be the graph obtained from  $D$  by reversing the arcs of  $F$  in  $D$ , then  $D'$  is acyclic.*

We now introduce a definition which is useful for a lemma we prove later.

**Definition 2.2.** Let  $D_\sigma = (V_\sigma, A)$  be an ordered directed graph and let  $f = vu$  be a backward arc of  $D_\sigma$ . We call *certificate* of  $f$ , and denote it by  $c(f)$ , any directed path from  $u$  to  $v$  using only forward arcs in the span of  $f$  in  $D_\sigma$ .

Observe that such a directed path together with the backward arc  $f$  forms a directed cycle in  $D_\sigma$  whose only backward arc is  $f$ .

**Definition 2.3.** Let  $D_\sigma = (V_\sigma, A)$  be an ordered directed graph, and let  $F \subseteq A$  be a set of backward arcs of  $D_\sigma$ . We say that we can *certify*  $F$  whenever it is possible to find a set  $\mathcal{F} = \{c(f) : f \in F\}$  of arc-disjoint certificates for the arcs in  $F$ .

Let  $D_\sigma = (V_\sigma, A)$  be an ordered directed graph, and let  $F \subseteq A$  be a subset of backward arcs of  $D_\sigma$ . We say that we can certify the set  $F$  using *only arcs from*  $A' \subseteq A$  if  $F$  can be certified by a collection  $\mathcal{F}$  such that the union of the arcs of the certificates in  $\mathcal{F}$  is contained in  $A'$ . In the following,  $\text{fas}(D)$  denotes the size of a minimum feedback arc set, that is, the cardinality of a minimum sized set  $F$  of arcs whose removal makes  $D$  acyclic.

**Lemma 2.4.** *Let  $D_\sigma$  be an ordered directed graph, and let  $\mathcal{P} = \{V_1, \dots, V_l\}$  be a partition of  $D_\sigma$  into intervals. Assume that the set  $F$  of all backward arcs of  $D_\sigma[A_B]$  can be certified using only arcs from  $A_B$ . Then  $\text{fas}(D_\sigma) = \text{fas}(D_\sigma[A_I]) + \text{fas}(D_\sigma[A_B])$ . Moreover, there exists a minimum sized feedback arc set of  $D_\sigma$  containing  $F$ .*

**Proof.** For any bipartition of the arc set  $A$  into  $A_1$  and  $A_2$ ,  $\text{fas}(D_\sigma) \geq \text{fas}(D_\sigma[A_1]) + \text{fas}(D_\sigma[A_2])$ . Hence, in particular for a partition of the arc set  $A$  into  $A_I$  and  $A_B$  we have that  $\text{fas}(D_\sigma) \geq \text{fas}(D_\sigma[A_I]) + \text{fas}(D_\sigma[A_B])$ . Next, we show that  $\text{fas}(D_\sigma) \leq \text{fas}(D_\sigma[A_I]) + \text{fas}(D_\sigma[A_B])$ . This follows from the fact that once we reverse all the arcs in  $F$ , each remaining directed cycle lies in  $D_\sigma[V_i]$  for some  $i \in \{1, \dots, l\}$ . In other words once we reverse all the arcs in  $F$ , every cycle is completely contained in  $D_\sigma[A_i]$ . This concludes the proof of the first part of the lemma. In fact, what we have shown is that there exists a minimum sized feedback arc set of  $D_\sigma$  containing  $F$ . This concludes the proof of the lemma.  $\square$

### 3. Kernels for $k$ -FAST

In this section we first give a subquadratic vertex kernel of size  $O(k\sqrt{k})$  for  $k$ -FAST and then improve on it to get our final vertex kernel of size  $O(k)$ . We start by giving a few reduction rules that will be needed to bound the size of the kernels.

**Rule 3.1.** If a vertex  $v$  is not contained in any triangle, delete  $v$  from  $T$ .

**Rule 3.2.** If there exists an arc  $uv$  that belongs to more than  $k$  distinct triangles, then reverse  $uv$  and decrease  $k$  by 1.

We say that a reduction rule is *sound*, if whenever the rule is applied to an instance  $(T, k)$  to obtain an instance  $(T', k')$ ,  $T$  has a feedback arc set of size at most  $k$  if and only if  $T'$  has a feedback arc set of size at most  $k'$ . Moreover, applying a reduction rule in polynomial time means that the structure sought by the reduction rule can be identified in polynomial time and the instance can be updated in polynomial time. Finally, we say that an instance  $(T, k)$  is *reduced* according to a set of reduction rules whenever none of the reduction rules can be applied to  $(T, k)$ .

**Lemma 3.1.** (See [4,16].) *Rules 3.1 and 3.2 are sound and can be applied in polynomial time.*

Rules 3.1 and 3.2 together led to a quadratic kernel for  $k$ -WFAST [4]. Earlier, these rules were used by Dom et al. [16] to obtain a quadratic kernel for  $k$ -FAST. We now add a new reduction rule that will allow us to obtain the claimed bound on the kernel sizes for  $k$ -FAST. Given an ordered tournament  $T_\sigma = (V_\sigma, A)$ , we say that  $\mathcal{P} = \{V_1, \dots, V_l\}$  is a *safe partition* of  $V_\sigma$  into intervals whenever it is possible to certify the backward arcs of  $T_\sigma[A_B]$  using only arcs from  $A_B$ .

**Rule 3.3.** Let  $T_\sigma$  be an ordered tournament, and  $\mathcal{P} = \{V_1, \dots, V_l\}$  be a safe partition of  $V_\sigma$  into intervals such that  $F \neq \emptyset$ , where  $F$  denotes the set of backward arcs of  $T_\sigma[A_B]$ . Then reverse all the arcs of  $F$  and decrease  $k$  by  $|F|$ .

**Lemma 3.2.** *Rule 3.3 is sound.*

**Proof.** Let  $\mathcal{P}$  be a safe partition of  $T_\sigma$ . Observe that it is possible to certify all the backward arcs, that is  $F$ , using only arcs in  $A_B$ . Hence using Lemma 2.4 we have that  $\text{fas}(T_\sigma) = \text{fas}(T_\sigma[A_I]) + \text{fas}(T_\sigma[A_B])$ . Furthermore, by Lemma 2.4 we also know that there exists a minimum sized feedback arc set of  $D_\sigma$  containing  $F$ . Thus,  $T_\sigma$  has a feedback arc set of size at most  $k$  if and only if the tournament  $T'_\sigma$  obtained from  $T_\sigma$  by reversing all the arcs of  $F$  has a feedback arc set of size at most  $k - |F|$ .  $\square$

#### 3.1. A subquadratic kernel for $k$ -FAST

In this section, we show how to obtain an  $O(k\sqrt{k})$  sized vertex kernel for  $k$ -FAST. To do so, we introduce the following reduction rule (see Fig. 1).

**Rule 3.4.** Let  $T_m$  be a maximal transitive module of size  $p$ , and  $I$  and  $O$  be the set of in-neighbors and out-neighbors of the vertices of  $T_m$  in  $T$ , respectively. Let  $Z$  be the set of arcs  $uv$  such that  $u \in O$  and  $v \in I$ . If  $q = |Z| < p$  then reverse all the arcs in  $Z$  and decrease  $k$  by  $q$ .

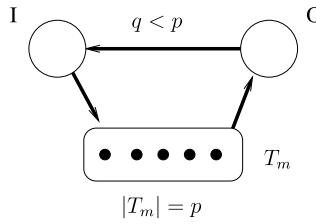


Fig. 1. A transitive module on which Rule 3.4 applies.

**Lemma 3.3.** Rule 3.4 is sound and can be applied in  $O(n + m)$  time.

**Proof.** We first prove that the partition  $\mathcal{P} = \{I, T_m, O\}$  forms a safe partition of the input tournament. Let  $T'_m = \{w_1, \dots, w_q\} \subseteq T_m$  be an arbitrary subset of size  $q$  of  $T_m$  and let  $Z = \{u_i v_i \mid 1 \leq i \leq q\}$ . Consider the collection  $\mathcal{F} = \{v_i w_i u_i \mid u_i v_i \in Z, w_i \in T'_m\}$  and notice that it certifies all the arcs in  $Z$ . In fact we have managed to certify all the backwards arcs of the partition using only arcs from  $A_B$  and hence  $\mathcal{P}$  forms a safe partition. Thus, by Rule 3.3, it is safe to reverse all the arcs from  $O$  to  $I$ .

The time complexity follows from the fact that computing the modular decomposition tree can be done in  $O(n + m)$  time on directed graphs [27]. It is well known that the modular decomposition tree of a tournament has nodes labelled either *prime* or *transitive* and that each maximal transitive module corresponds to the set of leaves attached to some transitive node of the modular decomposition tree.  $\square$

We show that any YES-instance to which none of Rules 3.1, 3.2 and 3.4 could be applied has at most  $O(k\sqrt{k})$  vertices.

**Theorem 3.4.** Let  $(T = (V, A), k)$  be a YES-instance to  $k$ -FAST which has been reduced according to Rules 3.1, 3.2 and 3.4. Then  $T$  has at most  $O(k\sqrt{k})$  vertices.

**Proof.** Let  $S$  be a feedback arc set of size at most  $k$  of  $T$  and let  $T'$  be the tournament obtained from  $T$  by reversing all the arcs in  $S$ . Let  $\sigma$  be the transitive ordering of  $T'$  and  $T_\sigma = (V_\sigma, A)$  be the ordered tournament corresponding to the ordering  $\sigma$ . We say that a vertex is *affected* if it is incident to some arc in  $S$ . Thus, the number of affected vertices is at most  $2|S| \leq 2k$ . The reduction Rule 3.1 ensures that the first and last vertex of  $T_\sigma$  are affected. To see this note that if the first vertex in  $V_\sigma$  is not affected then it is a source vertex (vertex with in-degree 0) and hence it is not part of any triangle and thus Rule 3.1 would have applied. We can similarly argue for the last vertex. Next we argue that there is no backward arc  $e$  of length greater than  $2k + 2$  in  $T_\sigma$ . Assume to the contrary that  $e = uv$  is a backward arc with  $S(e) = \{v, x_1, x_2, \dots, x_{2k+1}, \dots, u\}$  and hence  $l(e) > 2k + 2$ . Consider the collection  $\mathcal{T} = \{vx_i u \mid 1 \leq i \leq 2k\}$  and observe that at most  $k$  of these triples can contain an arc from  $S \setminus \{e\}$  and hence there exist at least  $k + 1$  triplets in  $\mathcal{T}$  which corresponds to distinct triangles all containing  $e$ . But then  $e$  would have been reversed by an application of Rule 3.2. Hence, we have shown that there is no backward arc  $e$  of length greater than  $2k + 2$  in  $T_\sigma$ . Thus  $\sum_{e \in S} l(e) \leq 2k^2 + 2k$ .

We also know that between two consecutive affected vertices there is exactly one maximal transitive module. Let us denote by  $t_i$  the number of vertices in these modules, where  $i \in \{1, \dots, 2k - 1\}$ . The objective here is to bound the number of vertices in  $V_\sigma$  or  $V$  using  $\sum_{i=1}^{2k-1} t_i$ . To do so, observe that since  $T$  is reduced under Rule 3.4, there are at least  $t_i$  backward arcs above every module with  $t_i$  vertices, each of length at least  $t_i$ . This implies that  $\sum_{i=1}^{2k-1} t_i^2 \leq \sum_{e \in S} l(e) \leq 2k^2 + 2k$ . Now, using the Cauchy–Schwarz inequality we can show the following:

$$\sum_{i=1}^{2k-1} t_i = \sum_{i=1}^{2k-1} t_i \cdot 1 \leq \sqrt{\sum_{i=1}^{2k-1} t_i^2 \cdot \sum_{i=1}^{2k-1} 1} \leq \sqrt{(2k^2 + 2k) \cdot (2k - 1)} = \sqrt{4k^3 + 2k^2 - 2k}.$$

Thus every reduced YES-instance has at most  $\sqrt{4k^3 + 2k^2 - 2k} + 2k = O(k\sqrt{k})$  vertices.  $\square$

### 3.2. A linear kernel for $k$ -FAST

We begin this subsection by showing some general properties about tournaments which will be useful in obtaining a linear kernel for  $k$ -FAST.

#### 3.2.1. Backward weighted tournaments

Let  $T_\sigma$  be an ordered tournament with weights on its backward arcs. We call such a tournament a *backward weighted tournament* and denote it by  $T_\omega$ , and use  $\omega(e)$  to denote the weight of a backward arc  $e$ . For every interval  $I := [v_i, \dots, v_j]$  we use  $\omega(I)$  to denote the total weight of all backward arcs having both their endpoints in  $I$ , that is,  $\omega(I) = \sum_{e=uv} \omega(e)$  where  $u, v \in I$  and  $e$  is a backward arc.

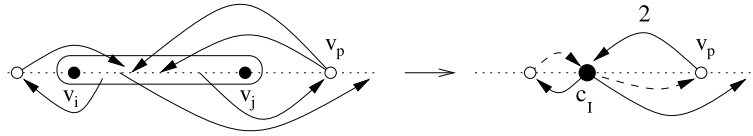


Fig. 2. Illustration of the contraction step for the interval  $I := [v_i, \dots, v_j]$ .

**Definition 3.5 (Contraction).** Let  $T_\omega = (V_\sigma, A)$  be an ordered tournament with weights on its backward arcs and  $I = [v_i, \dots, v_j]$  be an interval. The contracted tournament is defined as  $T_{\omega'} = (V_{\sigma'} = V_\sigma \setminus \{I\} \cup \{c_I\}, A')$ . The arc set  $A'$  is defined as follows.

- It contains all the arcs  $A_1 = \{uv \mid uv \in A, u \notin I, v \notin I\}$ .
- Add  $A_2 = \{uc_I \mid uv \in A, u \notin I, v \in I\}$  and  $A_3 = \{c_Iv \mid uv \in A, u \in I, v \notin I\}$ .
- Finally, we remove every forward arc involved in a 2-cycle after the addition of arcs in the previous step.

The order  $\sigma'$  for  $T_{\omega'}$  is provided by  $\sigma' = v_1, \dots, v_{i-1}, c_I, v_{j+1}, \dots, v_n$ . We define the weight of a backward arc  $e = xy$  of  $A'$  as follows:

$$w'(xy) = \begin{cases} w(xy) & \text{if } xy \in A_1, \\ \sum_{\{xz \in A \mid z \in I\}} w(xz) & \text{if } xy \in A_2, \\ \sum_{\{zy \in A \mid z \in I\}} w(zy) & \text{if } xy \in A_3. \end{cases}$$

We refer to Fig. 2 for an illustration.

Next we generalize the notions of certificate and certification (Definitions 2.2 and 2.3) to backward weighted tournaments.

**Definition 3.6.** Let  $T_\omega = (V_\sigma, A)$  be a backward weighted tournament, and let  $f = vu \in A$  be a backward arc of  $T_\omega$ . We call  $\omega$ -certificate of  $f$ , and denote it by  $\mathcal{C}(f)$ , a collection of  $\omega(f)$  arc-disjoint directed paths going from  $u$  to  $v$  and using only forward arcs in the span of  $f$  in  $T_\omega$ .

**Definition 3.7.** Let  $T_\omega = (V_\sigma, A)$  be a backward weighted tournament, and let  $F \subseteq A$  be a subset of backward arcs of  $T_\omega$ . We say that we can  $\omega$ -certify  $F$  whenever it is possible to find a set  $\mathcal{F} = \{\mathcal{C}(f) : f \in F\}$  of arc-disjoint  $\omega$ -certificates for the arcs in  $F$ .

**Lemma 3.8.** Let  $T_\omega = (V_\sigma, A)$  be a backward weighted tournament such that for every interval  $I := [v_i, \dots, v_j]$  the following holds:

$$2 \cdot \omega(I) \leq |I| - 1. \tag{1}$$

Then it is possible to  $\omega$ -certify the backward arcs of  $T_\omega$ .

**Proof.** Let  $V_\sigma = v_1, \dots, v_n$ . The proof is by induction on  $n$ , the number of vertices. Note that by applying (1) to the interval  $I = [v_1, \dots, v_n]$ , we have that there exists a vertex  $v_i$  in  $T_\omega$  that is not incident to any backward arc. Let  $T'_\omega = (V'_\sigma, A')$  denote the tournament  $T_\omega \setminus \{v_i\}$ . We say that an interval  $I$  is critical whenever  $|I| \geq 2$  and  $2 \cdot \omega(I) = |I| - 1$ . We now consider several cases, based on different types of critical intervals.

- Suppose that there are no critical intervals. Thus, in  $T'_\omega$ , every interval satisfies (1), and hence by induction on  $n$  the result holds.
- Suppose now that the only critical interval is  $I = [v_1, \dots, v_n]$ , and let  $e = vu$  be a backward arc above  $v_i$  with the maximum length. Note that since  $v_i$  does not belong to any backward arc, we can use it to form a directed path  $c(e) = uv_i v$ , which is a certificate for  $e$ . We now consider  $T'_\omega$  where the weight of  $e$  has been decreased by 1. In this process if  $\omega(e)$  becomes 0 then we reverse the arc  $e$ . We now show that every interval of  $T'_\omega$  respects (1). If an interval  $I' \in T'_\omega$  does not contain  $v_i$  in the corresponding interval in  $T_\omega$ , then by our assumption we have that  $2 \cdot \omega(I') \leq |I'| - 1$ . Now we assume that the interval corresponding to  $I'$  in  $T_\omega$  contains  $v_i$  but either  $u \notin I' \cup \{v_i\}$  or  $v \notin I' \cup \{v_i\}$ . Then we have  $2 \cdot \omega(I') = 2 \cdot \omega(I) < |I| - 1 = |I'|$  and hence we get that  $2 \cdot \omega(I') \leq |I'| - 1$ . Finally, we assume that the interval corresponding to  $I'$  in  $T_\omega$  contains  $v_i$  and  $u, v \in I' \cup \{v_i\}$ . In this case,  $2 \cdot \omega(I') = 2 \cdot (\omega(I) - 1) \leq |I| - 1 - 2 < |I'| - 1$ . Thus, by the induction hypothesis, we obtain a family of arc-disjoint  $\omega$ -certificates  $\mathcal{F}'$  which  $\omega$ -certify the backward arcs of  $T'_\omega$ . Observe that the maximality of  $l(e)$  ensures that if  $e$  is reversed then it will not be used in any  $\omega$ -certificate of  $\mathcal{F}'$ , thus implying that  $\mathcal{F}' \cup c(e)$  is a family  $\omega$ -certifying the backward arcs of  $T_\omega$ .
- Finally, suppose that there exists a critical interval  $I \subsetneq V_\sigma$ . Roughly speaking, we will show that  $I$  and  $V_\sigma \setminus I$  can be certified separately. To do so, we first show the following.

**Claim 1.** Let  $I \subset V_\sigma$  be a critical interval. Then the tournament  $T_{\omega'} = (V_{\sigma'}, A')$  obtained from  $T_\omega$  by contracting  $I$  satisfies the conditions of the lemma.

**Proof.** Let  $H'$  be any interval of  $T_{\omega'}$ . As before if  $H'$  does not contain  $c_I$  then the result holds by hypothesis. Otherwise, let  $H$  be the interval corresponding to  $H'$  in  $T_\omega$ . We will show that  $2\omega(H') \leq |H'| - 1$ . By hypothesis, we know that  $2\omega(H) \leq |H| - 1$  and that  $2\omega(I) = |I| - 1$ . Thus we have the following.

$$2\omega(H') = 2 \cdot (\omega(H) - \omega(I)) \leq |H| - 1 - |I| + 1 = (|H| + 1 - |I|) - 1 = |H'| - 1.$$

Thus, we have shown that the tournament  $T_{\omega'}$  satisfies the conditions of the lemma.  $\square$

We now consider a minimal critical interval  $I$ . By induction, and using the claim, we know that we can obtain a family of arc-disjoint  $\omega$ -certificates  $\mathcal{F}'$  which  $\omega$ -certifies the backward arcs of  $T_{\omega'}$  without using any arc within  $I$ . Now, by minimality of  $I$ , we can use (ii) to obtain a family of arc-disjoint  $\omega$ -certificates  $\mathcal{F}''$  which  $\omega$ -certifies the backward arcs of  $I$  using only arcs within  $I$ . Thus,  $\mathcal{F}' \cup \mathcal{F}''$  is a family  $\omega$ -certifying all backward arcs of  $T_\omega$ .

This concludes the proof of the lemma.  $\square$

In the following, any interval that does not respect condition (1) is said to be a *dense interval*.

**Lemma 3.9.** Let  $T_\omega = (V_\sigma, A)$  be a backward weighted tournament reduced under Rule 3.1 with  $|V_\sigma| \geq 2p + 1$  and  $\omega(V_\sigma) \leq p$ . Then there exists a safe partition of  $V_\sigma$  with at least one backward arc between the intervals and it can be computed in polynomial time.

**Proof.** The proof is by induction on  $n = |V_\sigma|$ . Observe that by hypothesis every vertex of  $T_\omega$  belongs to the span of some backward arc (since otherwise it would not be contained in any triangle). It follows that the statement is true for  $n = 3$ : in such a case,  $T_\omega$  is a triangle with exactly one backward arc, and hence the partition into singletons is a safe partition. This constitutes our base case.

For the inductive step, we assume first that there is no dense interval in  $T_\omega$ . In this case Lemma 3.8 ensures that the partition of  $V_\sigma$  into singletons of vertices is a safe partition. So from now on we assume that there exists at least one dense interval.

Let  $I$  be a dense interval. By definition of  $I$ , we have that  $\omega(I) \geq \frac{1}{2} \cdot |I|$ . We now contract  $I$  and obtain the backward weighted tournament  $T_{\omega'} = (V_{\sigma'}, A')$ . In the contracted tournament  $T_{\omega'}$ , we have:

$$\begin{cases} |V_{\sigma'}| \geq 2p + 1 - (|I| - 1) = 2p - |I| + 2; \\ \omega'(V_{\sigma'}) \leq p - \frac{1}{2} \cdot |I|. \end{cases}$$

Thus, if we set  $r := p - \frac{1}{2} \cdot |I|$ , we get that  $|V_{\sigma'}| \geq 2r + 1$  and  $\omega'(V_{\sigma'}) \leq r$ . Since  $|V_{\sigma'}| < |V_\sigma|$ , by the induction hypothesis we can find a safe partition  $\mathcal{P}$  of  $T_{\omega'}$ , and thus obtain a family  $\mathcal{F}_{\omega'}$  that  $\omega$ -certifies the backward arcs of  $T_{\omega'}[A_B]$  using only arcs in  $A_B$ . Observe that every vertex of  $T_{\omega'}$  still belongs to the span of some backward arc, and hence it is safe to apply our induction hypothesis.

We claim that  $\mathcal{P}'$  obtained from  $\mathcal{P}$  by substituting  $c_I$  by its corresponding interval  $I$  is a safe partition in  $T_\omega$ . To see this, first observe that if  $c_I$  has not been used to  $\omega$ -certify the backward arcs in  $T_{\omega'}[A_B]$ , that is, if  $c_I$  is not an end point of any arc in the  $\omega$ -certificates, then we are done. So from now on we assume that  $c_I$  has been part of an  $\omega$ -certificate for some backward arc. Let  $e = vu$  be such a backward arc in  $T_{\omega'}[A_B]$ , and let  $c_{\omega'}(e) \in \mathcal{F}_{\omega'}$  be a  $\omega$ -certificate of  $e$ . First we assume that  $c_I$  is neither the first nor the last vertex of the certificate  $c_{\omega'}(e)$  (with respect to ordering  $\sigma'$ ), and let  $c_1$  and  $c_2$  be the left (in-) and right (out-) neighbors of  $c_I$  in  $c_{\omega'}(e)$ . By definition of the contraction step together with the fact that there is a forward arc between  $c_1$  and  $c_I$  and between  $c_I$  and  $c_2$  in  $T_{\omega'}$ , we have that there were no backward arcs between any vertex in the interval corresponding to  $c_I$  and  $c_1$  and  $c_2$  in the original tournament  $T_\omega$ . So we can always find a vertex in  $I$  to replace  $c_I$  in  $c_{\omega'}(e)$ , thus obtaining a certificate  $c(e)$  for  $e$  in  $T_\omega[A_B]$  (observe that  $e$  remains a backward arc even in  $T_\omega$ ). Now we assume that  $c_I$  is either the first or last vertex in the certificate  $c_{\omega'}(e)$ . Let  $e'$  be an arc in  $T_\omega$  corresponding to  $e$  in  $T_{\omega'}$  with one of its endpoints being  $e_I \in I$ . To certify  $e'$  in  $T_\omega[A_B]$ , we need to show that we can construct a certificate  $c(e')$  using only arcs of  $T_\omega[A_B]$ . We have two cases to deal with.

- (i) If  $c_I$  is the first vertex of  $c_{\omega'}(e)$ , then let  $c_1$  be the right neighbor of  $c_I$  for any directed path  $P$  between  $u = c_I$  and  $v$  in  $c_{\omega'}(e)$ . Using the same argument as before, there are only forward arcs between any vertex in  $I$  and  $c_1$ . In particular, there is a forward arc  $e_I c_1$  in  $T_\omega$ , meaning that we can construct a  $\omega$ -certificate for  $e'$  in  $T_\omega$  by setting  $c(e') := (c_{\omega'}(e) \setminus \{c_I\}) \cup \{e_I\}$ . (See Fig. 3.)
- (ii) If  $c_I$  is the last vertex of  $c_{\omega'}(e)$ , then let  $c_q$  be the left neighbor of  $c_I$  for any directed path  $P$  between  $u$  and  $v = c_I$  in  $c_{\omega'}(e)$ . Once again, we have that there are only forward arcs between  $c_q$  and vertices in  $I$ , and thus between  $c_q$  and  $e_I$ . So using this we can construct a  $\omega$ -certificate for  $e'$  in  $T_\omega$ .



Fig. 3. On the left, the  $\omega$ -certificate  $c_{\omega'}(e) \in \mathcal{F}_{\omega'}$ . On the right, the corresponding  $\omega$ -certificate obtained in  $T_{\omega}$  by replacing  $c_1$  by the interval  $I$ .

**Algorithm 1:** Kernelization algorithm for the  $(2 + \epsilon)k$ -vertex kernel of  $k$ -FAST.

```

Input: An instance  $T = ((V, A), k)$  of  $k$ -FAST, where  $k \in \mathbb{N}$ , and a fixed  $\epsilon > 0$ .
Output: An equivalent instance  $T' = ((V', A'), k')$  with  $|V'| \leq (2 + \epsilon)k$ .

1 Reduce  $(T, k)$  according to Rule 3.1;
2 Compute a feedback arc set  $S$  using the  $(1 + \frac{\epsilon}{2})$ -PTAS for  $k$ -FAST [26];
3 if  $(|S| > (1 + \frac{\epsilon}{2})k)$  then
4    $\perp$  Return a small trivial No-instance;
5 else
6    $T_{\sigma} \leftarrow$  the transitive ordering obtained by reversing every arc of  $S$  in  $T$  (observe that  $\omega(T_{\sigma}) \leq p := (1 + \frac{\epsilon}{2})k$ );
7   repeat
8     if  $|V_{\sigma}| \geq 2p + 1 > (2 + \epsilon)k$  then
9        $\mathcal{P} \leftarrow$  the safe partition (with at least one backward arc between its intervals) obtained by Lemma 3.9;
10       $T_{\sigma} \leftarrow$  reverse all backward arcs between intervals of  $\mathcal{P}$  in  $T_{\sigma}$  (Rule 3.3);
11       $k \leftarrow$  decrease the value of  $k$  accordingly;
12      if  $(k \geq 0)$  then
13        Reduce  $(T_{\sigma}, k)$  according to Rule 3.1;
14        if  $(V_{\sigma} = \emptyset)$  then
15           $\perp$  Return a small trivial Yes-instance;
16  until  $(k \leq 0)$  or  $(|V_{\sigma}| \leq (2 + \epsilon)k)$ ;
17  if  $(k \leq 0)$  then
18     $\perp$  Return a small trivial No-instance;
19  else
20     $\perp$  Return  $T_{\sigma}$ ;

```

Notice that the fact that all  $\omega$ -certificates are pairwise arc-disjoint in  $T_{\omega'}[A_B]$  implies that the corresponding  $\omega$ -certificates are arc-disjoint in  $T_{\omega}[A_B]$ , and so  $\mathcal{P}'$  is indeed a safe partition of  $V_{\sigma}$ .  $\square$

We are now ready to give the linear size kernel for  $k$ -FAST. To do so, we make use of the fact that there exists a polynomial time approximation scheme for this problem [26]. The kernelization algorithm is depicted in Algorithm 1.

**Theorem 3.10.** For every fixed  $\epsilon > 0$ , there exists a vertex kernel for  $k$ -FAST with at most  $(2 + \epsilon)k$  vertices that can be computed in polynomial time.

**Proof.** Let  $(T = (V, A), k)$  be an instance of  $k$ -FAST. First, we reduce  $(T, k)$  according to Rule 3.1. Then, for a fixed  $\epsilon > 0$ , we compute a feedback arc set  $S$  using the known  $(1 + \frac{\epsilon}{2})$ -polynomial time approximation scheme for  $k$ -FAST [26]. If  $|S| > (1 + \frac{\epsilon}{2})k$ , then there is no feedback arc set of size at most  $k$  for  $T$ . Hence we return a trivial small No-instance. Otherwise,  $S$  has size at most  $(1 + \frac{\epsilon}{2})k$ . We then order  $T$  with the transitive ordering of the tournament obtained by reversing every arc of  $S$  in  $T$ . Let  $T_{\sigma}$  denote the resulting ordered tournament. By the upper bound on the size of  $S$ , we know that  $T_{\sigma}$  has at most  $(1 + \frac{\epsilon}{2})k$  backward arcs. Thus, if  $T_{\sigma}$  has more than  $(2 + \epsilon)k$  vertices then Lemma 3.9 ensures that we can find a safe partition with at least one backward arc between the intervals in polynomial time. Hence we can reduce the tournament by applying Rule 3.3, decreasing the value of  $k$  accordingly. Finally, if  $k \geq 0$ , we reduce the tournament according to Rule 3.1; notice that if we get  $V = \emptyset$  doing so, then we return a small trivial Yes-instance. We repeat the previous steps until we do not find a safe partition or  $k \leq 0$ . In the former case, we know by Lemma 3.9 that  $T$  can have at most  $(2 + \epsilon)k$  vertices, thus implying the result. In the latter case we return a trivial small No-instance.  $\square$

**4. Conclusion**

In this paper we obtained linear vertex kernel for  $k$ -FAST, in fact, a vertex kernel of size  $(2 + \epsilon)k$  for any fixed  $\epsilon > 0$ . The new bound on the kernel size improves the previous known bound of  $O(k^2)$  on the vertex kernel size for  $k$ -FAST given in [4,16]. Moreover, it would be interesting to see if one can obtain kernels for other problems using either polynomial time approximation schemes or a constant factor approximation algorithm for the corresponding problem. An interesting problem which remains unanswered is, whether there exists a linear or even an  $o(k^2)$  vertex kernel for the  $k$ -FEEDBACK VERTEX SET IN TOURNAMENTS ( $k$ -FVST) problem. In the  $k$ -FVST problem we are given a tournament  $T$  and a positive integer  $k$

and the aim is to find a set of at most  $k$  vertices whose deletion makes the input tournament acyclic. The smallest known kernel for  $k$ -FVST has size  $O(k^2)$  [1].

## References

- [1] F.N. Abu-Khzam, A kernelization algorithm for d-hitting set, *J. Comput. System Sci.* 76 (7) (2010) 524–531.
- [2] N. Ailon, M. Charikar, A. Newman, Aggregating inconsistent information: ranking and clustering, in: *ACM Symposium on Theory of Computing (STOC)*, 2005, pp. 684–693.
- [3] N. Alon, Ranking tournaments, *SIAM J. Discrete Math.* 20 (1) (2006) 137–142.
- [4] N. Alon, D. Lokshtanov, S. Saurabh, Fast FAST, in: *ICALP*, in: *Lecture Notes in Comput. Sci.*, vol. 5555, Springer, 2009, pp. 49–58.
- [5] J. Bartholdi III, C.A. Tovey, M.A. Trick, Voting schemes for which it can be difficult to tell who won the election, *Soc. Choice Welf.* 6 (2) (1989) 157–165.
- [6] H.L. Bodlaender, R.G. Downey, M.R. Fellows, D. Hermelin, On problems without polynomial kernels, *J. Comput. System Sci.* 75 (8) (2009) 423–434.
- [7] H.L. Bodlaender, F.V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, D.M. Thilikos, (Meta) Kernelization, in: *FOCS*, 2009, pp. 629–638.
- [8] J. Borda, Mémoire sur les élections au scrutin, *Histoire de l'Académie Royale des Sciences*, 1781.
- [9] N. Bousquet, J. Daligault, S. Thomassé, A. Yeo, A polynomial kernel for multicut in trees, in: *Symposium on Theoretical Aspects of Computer Science (STACS)*, 2009, pp. 183–194.
- [10] P. Charbit, S. Thomassé, A. Yeo, The minimum feedback arc set problem is NP-hard for tournaments, *Combin. Probab. Comput.* 16 (1) (2007) 1–4.
- [11] I. Charon, O. Hudry, A survey on the linear ordering problem for weighted or unweighted tournaments, *4OR* 5 (1) (2007) 5–60.
- [12] W.W. Cohen, R.E. Schapire, Y. Singer, Learning to order things, in: *Advances in Neural Information Processing Systems (NIPS)*, 1997, pp. 451–457.
- [13] M. Condorcet, Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix, 1785.
- [14] D. Coppersmith, L. Fleischer, A. Rudra, Ordering by weighted number of wins gives a good ranking for weighted tournaments, in: *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006, pp. 776–782.
- [15] H. Dell, D. van Melkebeek, Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses, in: *STOC*, ACM, 2010, pp. 251–260.
- [16] M. Dom, J. Guo, F. Hüffner, R. Niedermeier, A. Truß, Fixed-parameter tractability results for feedback set problems in tournaments, in: *Conference on Algorithms and Complexity (CIAC)*, in: *Lecture Notes in Comput. Sci.*, vol. 3998, 2006, pp. 320–331.
- [17] M. Dom, D. Lokshtanov, S. Saurabh, Incompressibility through colors and IDs, in: *ICALP*, in: *Lecture Notes in Comput. Sci.*, vol. 5555, Springer, 2009, pp. 378–389.
- [18] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation revisited, Technical report.
- [19] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in: *World Wide Web Conference (WWW)*, 2001.
- [20] P. Erdős, J.W. Moon, On sets on consistent arcs in tournaments, *Canad. Math. Bull.* 8 (1965) 269–271.
- [21] F.V. Fomin, D. Lokshtanov, V. Raman, S. Saurabh, Fast local search algorithm for weighted feedback arc set in tournaments, in: *AAAI*, 2010, pp. 65–70.
- [22] H.A. Jung, On subgraphs without cycles in tournaments, *Combinatorial Theory and Its Applications II*, 1970, pp. 675–677.
- [23] M. Karpinski, W. Schudy, Faster algorithms for feedback arc set tournament, Kemeny rank aggregation and betweenness tournament, CoRR, arXiv:1006.4396, 2010.
- [24] J. Kemeny, *Mathematics without numbers*, Daedalus 88 (1959) 571–591.
- [25] J. Kemeny, J. Snell, *Mathematical Models in the Social Sciences*, Blaisdell, 1962.
- [26] C. Kenyon-Mathieu, W. Schudy, How to rank with few errors, in: *ACM Symposium on Theory of Computing (STOC)*, 2007, pp. 95–103.
- [27] R.M. McConnell, F. de Montgolfier, Linear-time modular decomposition of directed graphs, *Discrete Appl. Math.* 145 (2) (2005) 198–209.
- [28] V. Raman, S. Saurabh, Parameterized algorithms for feedback set problems and their duals in tournaments, *Theoret. Comput. Sci.* 351 (3) (2006) 446–458.
- [29] K.D. Reid, E.T. Parker, Disproof of a conjecture of Erdős and Moser on tournaments, *J. Combin. Theory* 9 (1970) 225–238.
- [30] S. Seshu, M.B. Reed, *Linear Graphs and Electrical Networks*, Addison-Wesley, 1961.
- [31] P. Slater, Inconsistencies in a schedule of paired comparisons, *Biometrika* 48 (1961) 303–312.
- [32] E. Speckenmeyer, On feedback problems in digraphs, in: *Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, in: *Lecture Notes in Comput. Sci.*, vol. 411, Springer, 1989, pp. 218–231.
- [33] J. Spencer, Optimal ranking of tournaments, *Networks* 1 (1971) 135–138.
- [34] S. Thomassé, A  $4k^2$  kernel for feedback vertex set, *ACM Trans. Algorithms* 6 (2) (2010).
- [35] A. van Zuylen, Deterministic approximation algorithms for ranking and clusterings, Technical Report 1431, Cornell ORIE, 2005.
- [36] A. van Zuylen, R. Hegde, K. Jain, D.P. Williamson, Deterministic pivoting algorithms for constrained ranking and clustering problems, in: *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007, pp. 405–414.
- [37] D.H. Younger, Minimum feedback arc sets for a directed graph, *IEEE Trans. Circuit Theory* 10 (1963) 238–245.