Fast FAST

Noga Alon[1⋆] Daniel Lokshtanov[2] Saket Saurabh[2]

[1] Tel Aviv University, Tel Aviv 69978, Israel and IAS, Princeton, NJ, 08540, USA.
nogaa@tau.ac.il
[2] Department of Informatics, University of Bergen, N-5020 Bergen, Norway.
{daniello|saket.saurabh}@ii.uib.no

Abstract. We present a randomized subexponential time, polynomial space parameterized algorithm for the k-Weighted Feedback Arc Set in Tournaments (k-FAST) problem. We also show that our algorithm can be derandomized by slightly increasing the running time. To derandomize our algorithm we construct a new kind of universal hash functions, that we coin *universal coloring families.* For integers m, k and r, a family $\mathcal{F}$ of functions from $[m]$ to $[r]$ is called a universal (m, k, r)-coloring family if for any graph G on the set of vertices $[m]$ with at most k edges, there exists an $f \in \mathcal{F}$ which is a proper vertex coloring of G. Our algorithm is the first non-trivial subexponential time parameterized algorithm outside the framework of bidimensionality.

1 Introduction

In a competition where everyone plays against everyone it is uncommon that the results are acyclic and hence one cannot rank the players by simply using a topological ordering. A natural ranking is one that minimizes the number of upsets, where an upset is a pair of players such that the lower ranked player beats the higher ranked one. The problem of finding such a ranking given the match outcomes is the Feedback Arc Set problem restricted to tournaments.

A *tournament* is a directed graph where every pair of vertices is connected by exactly one arc, and a *feedback arc set* is a set of arcs whose removal makes the graph acyclic. Feedback arc sets in tournaments are well studied, both from the combinatorial [16, 17, 20–22, 28, 31, 32, 35], statistical [29] and algorithmic [1, 2, 9, 25, 33, 34] points of view. The problem has several applications - in psychology it occurs in relation to *ranking by paired comparisons*: here you wish to rank some items by an objective, but you don't have access to the objective function, only to pairwise comparisons of the objects in question. An example for this setting is measuring people's preferences for food. The weighted generalization of the problem, Weighted Feedback Arc Set in Tournaments is applied in *rank aggregation*: Here we are given several rankings of a set of object s, and we wish to produce a single ranking

⋆ Research supported in part by a USA Israeli BSF grant, by a grant from the Israel Science Foundation, by an ERC advanced grant, by NSF grant CCF 0832797 and by the Ambrose Monell Foundation.

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 [5, 8], machine learning [7], and search engine ranking [14, 15]. A natural consistency measure for rank aggregation is the number of pairs that occur in different order in the two rankings. This leads to *Kemeney-Young rank aggregation* [23, 24], a special case of WEIGHTED FEEDBACK ARC SET IN TOURNAMENTS.

Unfortunately, the problem of finding a feedback arc set of minimum size in an unweighted tournament is NP-hard [2]. However, even the weighted version of the problem admits a polynomial time approximation scheme [25] and has been shown to be fixed parameter tractable [27]. One should note that the weighted generalization shown to admit a PTAS in [25] differs slightly from the one considered in this paper. We consider the following problem:

> k-WEIGHTED FEEDBACK ARC SET IN TOURNAMENTS (k-FAST)
> INSTANCE: A tournament $T = (V, A)$, a weight function $w : A \to \{x \in \mathbb{R} : x \geq 1\}$ and an integer k.
> QUESTION: Is there an arc set $S \subseteq A$ such that $\sum_{e \in S} w(e) \leq k$ and $T \setminus S$ is acyclic?

The fastest previously known parameterized algorithm for k-FAST by Raman and Saurabh [27] runs in time $O(2.415^k \cdot k^{4.752} + n^{O(1)})$, and it was an open problem of Guo et al. [19] whether k-FAST can be solved in time $2^k \cdot n^{O(1)}$. We give a randomized and a deterministic algorithm both running in time $2^{O(\sqrt{k} \log^2 k)} + n^{O(1)}$. Our algorithms run in subexponential time, a trait uncommon to parameterized algorithms. In fact, to the authors best knowledge the only parameterized problems for which non-trivial subexponential time algorithms are known are *bidimensional* problems in planar graphs or graphs excluding a certain fixed graph H as a minor [10, 11, 13].

Our randomized algorithm is based on a novel version of the color coding technique initiated in [4] combined with a divide and conquer algorithm and a k^2 kernel for the problem, due to Dom et al. [12]. In order to derandomize our algorithm we construct a new kind of universal hash functions, that we coin *universal coloring families.* For integers m, k and r, a family $\mathcal{F}$ of functions from $[m]$ to $[r]$ is called a universal (m, k, r)-coloring family if for any graph G on the set of vertices $[m]$ with at most k edges, there exists an $f \in \mathcal{F}$ which is a proper vertex coloring of G. In the last section of the paper we give an explicit construction of a $(10k^2, k, O(\sqrt{k}))$-coloring family $\mathcal{F}$ of size $|\mathcal{F}| \leq 2^{\tilde{O}(\sqrt{k})}$ and an explicit universal $(n, k, O(\sqrt{k}))$-coloring family $\mathcal{F}$ of size $|\mathcal{F}| \leq 2^{\tilde{O}(\sqrt{k})} \log n$. We believe that these constructions can turn out to be useful to solve other edge subset problems in dense graphs.

2 Preliminaries

For an arc weighted tournament we define the weight function $w^* : V \times V \to \mathbb{R}$ such that $w^*(u,v) = w(uv)$ if $uv \in A$ and 0 otherwise. Given a directed graph $D = (V, A)$ and a set F of arcs in A define $D\{F\}$ to be the directed graph obtained from D by reversing all arcs of F. In our arguments we will need the following characterization of minimal feedback arc sets in directed graphs.

Proposition 1. *Let $D = (V, A)$ be a directed graph and F be a subset of A. Then F is a minimal feedback arc set of D if and only if F is a minimal set of arcs such that $D\{F\}$ is a directed acyclic graph.*

Given a minimal feedback arc set F of a tournament T, the ordering σ corresponding to F is the unique topological ordering of $T\{F\}$. Conversely, given an ordering σ of the vertices of T, the feedback arc set F corresponding to σ is the set of arcs whose endpoint appears before their startpoint in σ. The cost of an arc set F is $\sum_{e \in F} w(e)$ and the cost of a vertex ordering σ is the cost of the feedback arc set corresponding to σ.

For a pair of integer row vectors $\hat{p} = [p_1, \ldots, p_t]$, $\hat{q} = [q_1, \ldots, q_t]$ we say that $\hat{p} \leq \hat{q}$ if $p_i \leq q_i$ for all i. The transpose of a row vector $\hat{p}$ is denoted by $\hat{p}^\dagger$. The t-sized vector $\hat{e}$ is $[1, 1, \ldots, 1]$, $\hat{0}$ is $[0, 0, \ldots, 0]$ and $\hat{e}_i$ is the t-sized vector with all entries 0 except for the i'th which is 1. Let $\tilde{O}(\sqrt{k})$ denote, as usual, any function which is $O(\sqrt{k}(\log k)^{O(1)})$. For any positive integer m put $[m] = \{1, 2, \ldots, m\}$.

3 Color and Conquer

Our algorithm consists of three steps. In the first step we reduce the instance to a problem kernel with at most $O(k^2)$ vertices, showing how to efficiently reduce the input tournament into one with $O(k^2)$ vertices, so that the original tournament has a feedback arc set of weight at most k, if and only if the new one has such a set. In the second step we randomly color the vertices of our graph with $t = \sqrt{8k}$ colors, and define the arc set A_c to be the set of arcs whose endpoints have different colors. In the last step the algorithm checks whether there is a weight k feedback arc set $S \subseteq A_c$. A summary of the algorithm is given in Figure 1.

3.1 Kernelization

For the first step of the algorithm we use the kernelization algorithm provided by Dom et al. [12]. They only show that the data reduction is feasible for the unweighted case, while in fact, it works for the weighted case as well. For completeness we provide a short proof of this. A triangle in what follows means a directed cyclic triangle.

1. Perform a data reduction to obtain a tournament T' of size $O(k^2)$.
2. Let $t = \sqrt{8k}$. Color the vertices of T' uniformly at random with colors from $\{1, \ldots, t\}$.
3. Let A_c be the set of arcs whose endpoints have different colors. Find a minimum weight feedback arc set contained in A_c, or conclude that no such feedback arc set exists.

Fig. 1. Outline of the algorithm for k-FAST.

Lemma 1. k-FAST *has a kernel with* $O(k^2)$ *vertices.*

Proof. We give two simple reduction rules.

1. If an arc e is contained in at least $k + 1$ triangles reverse the arc and reduce k by $w(e)$.
2. If a vertex v is not contained in any triangle, delete v from T.

The first rule is safe because any feedback arc set that does not contain the arc e must contain at least one arc from each of the $k + 1$ triangles containing e and thus must have weight at least $k + 1$. The second rule is safe because the fact that v is not contained in any triangle implies that all arcs between $N^-(v)$ and $N^+(v)$ are oriented from $N^-(v)$ to $N^+(v)$. Hence for any feedback arc set S_1 of $T[N^-(v)]$ and feedback arc set S_2 of $T[N^+(v)]$, $S_1 \cup S_2$ is a feedback arc set of T.

Finally we show that any reduced yes instance T has at most $k(k + 2)$ vertices. Let S be a feedback arc set of T with weight at most k. The set S contains at most k arcs, and for every arc $e \in S$, aside from the two endpoints of e, there are at most k vertices that are contained in a triangle containing e, because otherwise the first rule would have applied. Since every triangle in T contains an arc of S and every vertex of T is in a triangle, T has at most $k(k + 2)$ vertices. □

3.2 Probability of a Good Coloring

We now proceed to analyze the second step of the algorithm. What we aim for, is to show that if T does have a feedback arc set S of weight at most k, then the probability that S is a subset of A_c is at least $2^{-c\sqrt{k}}$ for some fixed constant c. We show this by showing that if we randomly color the vertices of a k edge graph G with $t = \sqrt{8k}$ colors, then the probability that G has been properly colored is at least $2^{-c\sqrt{k}}$.

Lemma 2. *If a graph on q edges is colored randomly with $\sqrt{8q}$ colors then the probability that G is properly colored is at least $(2e)^{-\sqrt{q/8}}$.*

Proof. Arrange the vertices of the graph by repeatedly removing a vertex of lowest degree. Let $d_1, d_2, \ldots, d_s$ be the degrees of the vertices when they have been removed. Then for each i, $d_i(s - i + 1) \leq 2q$, since when vertex i is removed each vertex had degree at least d_i. Furthermore, $d_i \leq s - i$ for all i, since the degree of the vertex removed can not exceed the number of remaining vertices at that point. Thus $d_i \leq \sqrt{2q}$ for all i. In the coloring, consider the colors of each vertex one by one starting from the last one, that is vertex number s. When vertex number i is colored, the probability that it will be colored by a color that differs from all those of its d_i neighbors following it is at least $(1 - \frac{d_i}{\sqrt{8q}}) \geq (2e)^{-d_i/\sqrt{8q}}$ because $\sqrt{8q} \geq 2d_i$. Hence the probability that G is properly colored is at least

$$\prod_{i=1}^{s}(1 - \frac{d_i}{\sqrt{8q}}) \geq \prod_{i=1}^{s}(2e)^{-d_i/\sqrt{8q}} = (2e)^{-\sqrt{q/8}}.$$

□

3.3 Solving a Colored Instance

Given a t-colored tournament T, we will say that an arc set F is colorful if no arc in F is monochromatic. An ordering σ of T is colorful if the feedback arc set corresponding to σ is colorful. An optimal colorful ordering of T is a colorful ordering of T with minimum cost among all colorful orderings. We now give an algorithm that takes a t-colored arc weighted tournament T as input and finds a colorful feedback arc set of minimum weight, or concludes that no such feedback arc set exists.

Observation 1 *Let $T = (V_1 \cup V_2 \cup \ldots \cup V_t, A)$ be a t-colored tournament. There exists a colorful feedback arc set of T if and only if $T[V_i]$ induces an acyclic tournament for every i.*

We say that a colored tournament T is *feasible* if $T[V_i]$ induces an acyclic tournament for every i. Let $n_i = |V_i|$ for every i and let $\hat{n}$ be the vector $[n_1, n_2 \ldots n_t]$. Let $\sigma = v_1 v_2 \ldots v_n$ be the ordering of V corresponding to a colorful feedback arc set F of T. For every color class V_i of T, let $v_i^1 v_i^2 \ldots v_i^{n_i}$ be the order in which the vertices of V_i appear according to σ. Observe that since F is colorful, $v_i^1 v_i^2 \ldots v_i^{n_i}$ must be the unique topological ordering of $T[V_i]$. We exploit this to give a dynamic programming algorithm for the problem.

Lemma 3. *Given a feasible t-colored tournament T, we can find a minimum weight colorful feedback arc set in $O(t \cdot n^{t+1})$ time and $O(n^t)$ space.*

Proof. For an integer $x \geq 1$, define $S_x = \{v_1 \ldots, v_x\}$ and $S_x^i = \{v_1^i \ldots, v_x^i\}$. Let $S_0 = S_0^i = \emptyset$. Notice that for any x there must be some x' such that $S_x \cap V_i = S_{x'}$. Given an integer vector $\hat{p}$ of length t in which the ith entry is between 0 and n_i, let $T(\hat{p})$ be $T[S_{p_1}^1 \cup S_{p_2}^2 \ldots \cup S_{p_t}^t]$.

For a feasible t-colored tournament T, let $\textsc{Fas}(T)$ be the weight of the minimum weight colorful feedback arc set of T. Observe that if a t-colored tournament T is feasible then so are all induced subtournaments of T, and hence the function $\textsc{Fas}$ is well defined on all induced subtournaments of T. We proceed to prove that the following recurrence holds for $\textsc{Fas}(T(\hat{p}))$.

$$\textsc{Fas}(T(\hat{p})) = \min_{i \,:\, \hat{p}_i > 0} \Big(\textsc{Fas}(T(\hat{p} - \hat{e}_i)) + \sum_{u \in V(T(\hat{p}))} w^*(v_{\hat{p}_i}^i, u)\Big) \tag{1}$$

First we prove that the left hand side is at most the right hand side. Let i be the integer that minimizes the right hand side. Taking the optimal ordering of $T(\hat{p} - \hat{e}_i)$ and appending it with $v_{\hat{p}_i}^i$ gives an ordering of $T(\hat{p})$ with cost at most $\textsc{Fas}(T(\hat{p} - \hat{e}_i)) + \sum_{u \in V(T(\hat{p}))} w^*(v_{\hat{p}_i}^i, u)$.

To prove that the right hand side is at most the left hand side, take an optimal colorful ordering σ of $T(\hat{p})$ and let v be the last vertex of this ordering. There is an i such that $v = v_{\hat{p}_i}^i$. Thus σ restricted to $V(T(\hat{p} - \hat{e}_i))$ is a colorful ordering of $T(\hat{p} - \hat{e}_i)$ and the total weight of the edges with startpoint in v and endpoint in $V(T(\hat{p} - \hat{e}_i))$ is exactly $\sum_{u \in V(T(\hat{p}))} w^*(v_{\hat{p}_i}^i, u)$. Thus the cost of σ is at least the value of the right hand side of the inequality, completing the proof.

Recurrence 1 naturally leads to a dynamic programming algorithm for the problem. We build a table containing $\textsc{Fas}(T(\hat{p}))$ for every $\hat{p}$. There are $O(n^t)$ table entries, for each entry it takes $O(nt)$ time to compute it giving the $O(t \cdot n^{t+1})$ time bound. $\square$

In fact, the algorithm provided in Lemma 3 can be made to run slightly faster by pre-computing the value of $\sum_{u \in V(T(\hat{p}))} w^*(v_{\hat{p}_i}^i, u))$ for every $\hat{p}$ and i using dynamic programming, and storing it in a table. This would let us reduce the time to compute a table entry using Recurrence 1 from $O(nt)$ to $O(t)$ yielding an algorithm that runs in time and space $O(t \cdot n^t)$.

Lemma 4. k-FAST *(for a tournament of size $O(k^2)$) can be solved in expected time $2^{O(\sqrt{k}\log k)}$ and $2^{O(\sqrt{k}\log k)}$ space.*

Proof. Our algorithm proceeds as described in Figure 1. The correctness of the algorithm follows from Lemma 3. Combining Lemmata 1, 2, 3 yields an expected running time of $O((2e)^{\sqrt{k/8}}) \cdot O(\sqrt{8k} \cdot (k^2 + 2k)^{1+\sqrt{8k}}) \leq 2^{O(\sqrt{k}\log k)}$ for finding a feedback arc set of weight at most k if one exists. The space required by the algorithm is $O((k^2 + 2k)^{1+\sqrt{8k}}) \leq 2^{O(\sqrt{k}\log k)}$. $\square$

The dynamic programming algorithm from Lemma 3 can be turned into a divide and conquer algorithm that runs in polynomial space, at a small cost in the running time.

Lemma 5. *Given a feasible t-colored tournament T, we can find a minimum weight colorful feedback arc set in time $O(n^{1+(t+2)\cdot\log n})$ in polynomial space.*

Proof. By expanding Recurrence (1) $\lfloor n/2 \rfloor$ times and simplifying the right hand side we obtain the following recurrence.

$$\textsc{Fas}(T(\hat{p})) = \min_{\substack{\hat{q} \geq \hat{0} \\ \hat{q}^{\dagger} \cdot \hat{e} = \lceil n/2 \rceil}} \{\textsc{Fas}(T(\hat{q})) + \textsc{Fas}(T \setminus V(T(\hat{q}))) + \sum_{\substack{u \in V(T(\hat{q})) \\ v \notin V(T(\hat{q}))}} w^*(v,u)\} \tag{2}$$

Recurrence 2 immediately yields a divide and conquer algorithm for the problem. Let $\mathcal{T}(n)$ be the running time of the algorithm restricted to a subtournament of T with n vertices. For a particular vector $\hat{q}$ it takes at most n^2 time to find the value of $\sum_{u \in V(T(\hat{q})), v \notin V(T(\hat{q}))} w^*(v,u)$. It follows that $\mathcal{T}(n) \leq n^{t+2} \cdot 2 \cdot \mathcal{T}(n/2) \leq 2^{\log n} \cdot n^{(t+2)\cdot\log n} = n^{1+(t+2)\cdot\log n}$. □

Theorem 1. k-FAST *(for a tournament of size $O(k^2)$) can be solved in expected time $2^{O(\sqrt{k}\log^2 k)}$ and polynomial space. Therefore,* k-FAST *for a tournament of size n can be solved in expected time $2^{O(\sqrt{k}\log^2 k)} + n^{O(1)}$ and polynomial space.*

4 Derandomization with Universal Coloring Families

For integers m, k and r, a family $\mathcal{F}$ of functions from $[m]$ to $[r]$ is called a universal (m, k, r)-coloring family if for any graph G on the set of vertices $[m]$ with at most k edges, there exists an $f \in \mathcal{F}$ which is a proper vertex coloring of G. An explicit construction of a $(10k^2, k, O(\sqrt{k}))$-coloring family can replace the randomized coloring step in the algorithm for k-FAST. In this section, we provide such a construction.

Theorem 2. *There exists an explicit universal $(10k^2, k, O(\sqrt{k}))$-coloring family $\mathcal{F}$ of size $|\mathcal{F}| \leq 2^{\tilde{O}(\sqrt{k})}$.*

For simplicity we omit all floor and ceiling signs whenever these are not crucial. We make no attempt to optimize the absolute constants in the $\tilde{O}(\sqrt{k})$ or in the $O(\sqrt{k})$ notation. Whenever this is needed, we assume that k is sufficiently large.

Proof. Let $\mathcal{G}$ be an explicit family of functions g from $[10k^2]$ to $[\sqrt{k}]$ so that every coordinate of g is uniformly distributed in $[\sqrt{k}]$, and every two coordinates are pairwise independent. There are known constructions of such a family $\mathcal{G}$ with $|\mathcal{G}| \leq k^{O(1)}$. Indeed, each function g represents the values of $10k^2$ pairwise independent random variables distributed uniformly in $[\sqrt{k}]$ in a point of a small sample space supporting such variables; a construction is given, for example, in [3]. The family $\mathcal{G}$ is obtained from the family of all linear polynomials over a finite field with some $k^{O(1)}$ elements, as described in [3].

We can now describe the required family $\mathcal{F}$. Each $f \in \mathcal{F}$ is described by a subset $T \subset [10k^2]$ of size $|T| = \sqrt{k}$ and by a function $g \in \mathcal{G}$. For each $i \in [10k^2]$, the value of $f(i)$ is determined as follows. Suppose $T = \{i_1, i_2, \ldots, i_{\sqrt{k}}\}$, with $i_1 < i_2 < \ldots < i_{\sqrt{k}}$. If $i = i_j \in T$, define $f(i) = \sqrt{k} + j$. Otherwise, $f(i) = g(i)$. Note that the range of f is of size $\sqrt{k} + \sqrt{k} = 2\sqrt{k}$, and the size of $\mathcal{F}$ is at most

$$\binom{10k^2}{\sqrt{k}}|\mathcal{G}| \leq \binom{10k^2}{\sqrt{k}}k^{O(1)} \leq 2^{O(\sqrt{k}\log k)} \leq 2^{\tilde{O}(\sqrt{k})}.$$

To complete the proof we have to show that for every graph G on the set of vertices $[10k^2]$ with at most k edges, there is an $f \in \mathcal{F}$ which is a proper vertex coloring of G. Fix such a graph G.

The idea is to choose T and g in the definition of the function f that will provide the required coloring for G as follows. The function g is chosen at random in $\mathcal{G}$, and is used to properly color all but at most $\sqrt{k}$ edges. The set T is chosen to contain at least one endpoint of each of these edges, and the vertices in the set T will be re-colored by a unique color that is used only once by f. Using the properties of $\mathcal{G}$ we now observe that with positive probability the number of edges of G which are monochromatic is bounded by $\sqrt{k}$.

Claim. If the vertices of G are colored by a function g chosen at random from $\mathcal{G}$, then the expected number of monochromatic edges is $\sqrt{k}$.

Proof. Fix an edge e in the graph G and $j \in [\sqrt{k}]$. As g maps the vertices in a pairwise independent manner, the probability that both the end points of e get mapped to j is precisely $\frac{1}{(\sqrt{k})^2}$. There are $\sqrt{k}$ possibilities for j and hence the probability that e is monochromatic is given by $\frac{\sqrt{k}}{(\sqrt{k})^2} = \frac{1}{\sqrt{k}}$. Let X be the random variable denoting the number of monochromatic edges. By linearity of expectation, the expected value of X is $k \cdot \frac{1}{\sqrt{k}} = \sqrt{k}$. □

Returning to the proof of the theorem, observe that by the above claim, with positive probability, the number of monochromatic edges is upper bounded by $\sqrt{k}$. Fix a $g \in \mathcal{G}$ for which this holds and let $T = \{i_1, i_2, \ldots, i_{\sqrt{k}}\}$ be a set of $\sqrt{k}$ vertices containing at least one endpoint of each monochromatic edge. Consider the function f defined by this T and g. As mentioned above f colors each of the vertices in T by a unique color, which is used only once by f, and hence we only need to consider the coloring of $G \setminus T$. However all edges in $G \setminus T$ are properly colored by g and f coincides with g on $G \setminus T$. Hence f is a proper coloring of G, completing the proof of the theorem. $\square$

Remarks:

– Each universal $(n, k, O(\sqrt{k}))$-coloring family must also be an $(n, \sqrt{k}, O(\sqrt{k}))$-hashing family, as it must contain, for every set S of $\sqrt{k}$ vertices in $[n]$, a function that maps the elements of S in a one-to-one manner, since these vertices may form a clique that has to be properly colored by a function of the family. Therefore, by the known bounds for families of hash functions (see, e.g., [26]), each such family must be of size at least $2^{\tilde{\Omega}(\sqrt{k})} \log n$.

Although the next result is not required for our results on the feedback arc set problem, we present it here as it may be useful in similar applications.

Theorem 3. *For any $n > 10k^2$ there exists an explicit universal $(n, k, O(\sqrt{k}))$-coloring family $\mathcal{F}$ of size $|\mathcal{F}| \leq 2^{\tilde{O}(\sqrt{k})} \log n$.*

Proof. Let $\mathcal{F}_1$ be an explicit $(n, 2k, 10k^2)$-family of hash functions from $[n]$ to $10k^2$ of size $|\mathcal{F}_1| \leq k^{O(1)} \log n$. This means that for every set $S \subset [n]$ of size at most $2k$ there is an $f \in \mathcal{F}_1$ mapping S in a one-to-one fashion. The existence of such a family is well known, and follows, for example, from constructions of small spaces supporting n nearly pairwise independent random variables taking values in $[10k^2]$. Let $\mathcal{F}_2$ be an explicit universal $(10k^2, k, O(\sqrt{k}))$-coloring family, as described in Theorem 2. The required family $\mathcal{F}$ is simply the family of all compositions of a function from $\mathcal{F}_2$ followed by one from $\mathcal{F}_1$. It is easy to check that $\mathcal{F}$ satisfies the assertion of Theorem 3. $\square$

Finally, combining the algorithm from Theorem 1 with the universal coloring family given by Theorem 2 yields a deterministic subexponential time polynomial space algorithm for k-FAST.

Theorem 4. k-FAST *can be solved in time $2^{\tilde{O}(\sqrt{k})} + n^{O(1)}$ and polynomial space.*

5 Concluding Remarks

In this article, we have shown that k-FAST can be solved in time $2^{\tilde{O}(\sqrt{k})} + n^{O(1)}$ and polynomial space. To achieve this we introduced a new variant of randomized color coding, and showed that this approach could be derandomized with an explicit construction of universal coloring families. We find it surprising that the problem admits a subexponential time parameterized algorithm, as even the existence of a $2^k \cdot n^{O(1)}$ time algorithm was an open problem until now.

At the end of the introduction of the paper in which it was proved that FEEDBACK ARC SET IN TOURNAMENTS admits a PTAS [25], Mathieu and Schudy write "*We can feel lucky that the FAS problem on tournaments turns out to be so easy as to have an approximation scheme: In contrast to Theorem* 1*, the related problem of feedback vertex set is hard to approximate even on tournaments.*" Interestingly, a similar remark can be made in our setting - a simple reduction from VERTEX COVER [30] shows that k-FEEDBACK VERTEX SET in tournaments can not be solved in subexponential time unless the Exponential Time Hypothesis [6, 18] fails.

The results of Section 4 can be extended to universal coloring families of uniform hypergraphs. These families can also be useful in tackling several parameterized algorithmic problems. The details will appear in the full version of this paper.

References

1. N. AILON, M. CHARIKAR, AND A. NEWMAN, *Aggregating inconsistent information: ranking and clustering*, in ACM Symposium on Theory of Computing (STOC), 2005, pp. 684–693.
2. N. ALON, *Ranking tournaments*, SIAM J. Discrete Math., 20 (2006), pp. 137–142.
3. N. ALON, L. BABAI, AND A. ITAI, *A fast and simple randomized parallel algorithm for the maximal independent set problem.*, Journal of Algorithms, 7 (1986), pp. 567–583.
4. N. ALON, R. YUSTER, AND U. ZWICK, *Color-coding*, J. Assoc. Comput. Mach., 42 (1995), pp. 844–856.
5. J. BORDA, *Mémoire sur les élections au scrutin*, Histoire de l'Académie Royale des Sciences, (1781).
6. L. CAI AND D. W. JUEDES, *On the existence of subexponential parameterized algorithms*, J. Comput. Syst. Sci., 67 (2003), pp. 789–807.
7. W. W. COHEN, R. E. SCHAPIRE, AND Y. SINGER, *Learning to order things*, in Advances in neural information processing systems (NIPS), 1997, pp. 451–457.
8. M. CONDORCET, *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*, 1785.
9. D. COPPERSMITH, L. FLEISCHER, AND 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.
10. E. D. DEMAINE, F. V. FOMIN, M. T. HAJIAGHAYI, AND D. M. THILIKOS, *Subexponential parameterized algorithms on bounded-genus graphs and -minor-free graphs*, J. ACM, 52 (2005), pp. 866–893.
11. E. D. DEMAINE AND M. HAJIAGHAYI, *The bidimensionality theory and its algorithmic applications*, Comput. J., 51 (2008), pp. 292–302.
12. M. DOM, J. GUO, F. HÜFFNER, R. NIEDERMEIER, AND A. TRUSS, *Fixed-parameter tractability results for feedback set problems in tournaments*, in CIAC, 2006, pp. 320–331.
13. F. DORN, F. V. FOMIN, AND D. M. THILIKOS, *Subexponential parameterized algorithms*, Computer Science Review, 2 (2008), pp. 29–39.
14. C. DWORK, R. KUMAR, M. NAOR, , AND D. SIVAKUMAR, *Rank aggregation revisited*, 2001.
15. C. DWORK, R. KUMAR, M. NAOR, AND D. SIVAKUMAR, *Rank aggregation methods for the web*, in WWW10, 2001.
16. P. ERDÖS AND J.W.MOON, *On sets on consistent arcs in tournaments*, Canadian Mathematical Bulletin, 8 (1965), pp. 269–271.

17. W. Fernandez de la Vega, *On the maximal cardinality of a consistent set of arcs in a random tournament*, J. Combinatorial Theory, Ser. B, 35 (1983), pp. 328–332.
18. J. Flum and M. Grohe, *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
19. J. Guo, H. Moser, and R. Niedermeier, *Iterative compression for exactly solving np-hard minimization problems.*, in Algorithmics of Large and Complex Networks, Lecture Notes in Computer Science, Springer, 2008, p. To appear.
20. H. Jung, *On subgraphs without cycles in tournaments*, Combinatorial Theory and its Applications II, (1970), pp. 675–677.
21. K.B.Reid, *On sets of arcs containing no cycles in tournaments*, Canad. Math Bulletin, 12 (1969), pp. 261–264.
22. K.D.Reid and E.T.Parker, *Disproof of a conjecture of erdos and moser on tournaments*, J. Combin. Theory, 9 (1970), pp. 225–238.
23. J. Kemeny, *Mathematics without numbers*, Daedalus, 88 (1959), pp. 571–591.
24. J. Kemeny and J. Snell, *Mathematical models in the social sciences*, Blaisdell, 1962.
25. C. Kenyon-Mathieu and W. Schudy, *How to rank with few errors.*, in STOC'07, ACM, 2007, pp. 95–103.
26. A. Nilli., *Perfect hashing and probability.*, Combinatorics, Probability and Computing, 3 (1994), pp. 407–409.
27. V. Raman and S. Saurabh, *Parameterized algorithms for feedback set problems and their duals in tournaments.*, Theoretical Computer Science, 351 (2006), pp. 446–458.
28. S. Seshu and M. Reed, *Linear Graphs and Electrical Networks*, Addison-Wesley, 1961.
29. P. Slater, *Inconsistencies in a schedule of paired comparisons*, Biometrika, 48 (1961), pp. 303–312.
30. E. Speckenmeyer, *On feedback problems in diagraphs*, in WG, 1989, pp. 218–231.
31. J. Spencer, *Optimal ranking of tournaments*, Networks, 1 (1971), pp. 135–138.
32. ———, *Optimal ranking of unrankable tournaments*, Period. Math. Hungar., 11 (1980), pp. 131–144.
33. A. van Zuylen, *Deterministic approximation algorithms for ranking and clusterings*, Tech. Rep. 1431, Cornell ORIE, 2005.
34. A. van Zuylen, R. Hegde, K. Jain, and D. P. Williamson, *Deterministic pivoting algorithms for constrained ranking and clustering problems*, in ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007, pp. 405–414.
35. D. Younger, *Minimum feedback arc sets for a directed graph*, IEEE Trans. Circuit Theory, 10 (1963), pp. 238–245.