Approximation Schemes for the Betweenness Problem in Tournaments and Related Ranking Problems

Marek Karpinski[*] Warren Schudy[†]

Abstract

We design the first *polynomial time approximation schemes (PTASs)* for the *Minimum Betweenness* problem in tournaments and some related higher arity ranking problems. This settles the approximation status of the Betweenness problem in tournaments along with other ranking problems which were open for some time now. We also show fixed parameter tractability of betweenness in tournaments and improved fixed parameter algorithms for feedback arc set tournament and Kemeny Rank Aggregation. The results depend on a new technique of dealing with fragile ranking constraints and could be of independent interest.

[*]Dept. of Computer Science and the Hausdorff Center for Mathematics, University of Bonn. Parts of this work done while visiting Microsoft Research. Email: marek@cs.uni-bonn.de

[†]Dept. of Computer Science, Brown University. Parts of this work done while visiting University of Bonn. Email: ws@cs.brown.edu

1 Introduction

We study the approximability of the Minimum Betweenness problem in tournaments (see [3]) that resisted so far efforts of designing polynomial time approximation algorithms with a constant approximation ratio. For the status of the general Betweenness problem, see e.g. [22, 13, 3, 12].

In this paper we design the first *polynomial time approximation scheme* (PTAS) for that problem, and generalize it to much more general class of ranking CSP problems, called here *fragile* problems. To our knowledge it is the first nontrivial approximation algorithm for the Betweenness problem in tournaments.

In the Betweenness problem we are given a ground set of *vertices* and a set of *betweenness constraints* involving 3 vertices and a *designated* vertex among them. The objective function of a ranking of the elements is the number of betweenness constraints for which the designated vertex is not between the other two vertices. The goal is to *minimize* the objective function. We refer to the Betweenness problem in tournaments, that in instances with a constraint for every triple of vertices, as the BetweennessTour or *fully dense* Betweenness problem (see [3]). We consider also the k-ary extension k-FAST of the Feedback-Arc-Set-Tournament (FAST) problem (see [20, 1, 4]).

We extend the above classes by introducing a more general class of *fragile ranking* k-CSP problems. A *constraint* S of a ranking k-CSP problem is called *fragile* if changing the relative order of a single vertex in S with respect to the rest of S makes it *unsatisfied* whenever S was satisfied by the original order. A *ranking* k-CSP problem is called *fragile* if all its constraints are fragile.

We now formulate our main results.

Theorem 1. *There exists a PTAS for the* BetweennessTour *problem.*

The above answers an open problem of [3] on the approximation status of the Betweenness problem in tournaments.

We now formulate our first generalization.

Theorem 2. *There exist PTASs for all* fragile ranking *k-CSP problems in tournaments.*

Theorem 2 entails, among other things, existence of a PTAS for the k-ary extension of FAST.

Corollary 1. *There exists a PTAS for the k-*FAST *problem.*

We generalize BetweennessTour to arities $k \geq 4$ by specifying for each constraint S a pair of vertices in S that must be placed at the ends of the ranking induced by the vertices in S. Such constraints do not satisfy our definition of fragile, but do satisfy a weaker notion that we call *weak fragility.* The definition of weakly fragile is identical to the definition for fragile except that only four particular single vertex moves are considered, namely swapping the first two vertices, swapping the last two, and moving the first or last vertex to the other end. We now formulate our most general theorem.

Theorem 3. *There exist PTASs for all* weak-fragile ranking *k-CSP problems in tournaments.*

Corollary 2. *There exists a PTAS for the k-*BetweennessTour *problem.*

As an additional application of our techniques we improve the parameterized time complexity of several ranking problems.

Theorem 4. *There exists a parameterized subexponential algorithm for* FAST *with runtime $2^{O(\sqrt{K})} + n^{O(1)}$ for $OPT \leq K$. A variant of the algorithm uses $2^{O(\sqrt{K}\log K)} + n^{O(1)}$ time and $n^{O(1)}$ space.*

Both results in Theorem 4 improve the best up to now known parameterized runtime bound of Alon, Lokshtanov and Saurabh [6] for the feedback arc set tournament problem by a $\Theta(\log K)$ factor in the exponent. We also give improved results for the closely related problem of *Kemeny rank aggregation (KRA)*; see e.g. [2, 20].

Theorem 5. *Let m be the number of input rankings (voters), n the number of candidates, and $OPT \leq m\binom{n}{2}$ the (unscaled) optimum value. There exists a parameterized subexponential algorithm for Kemeny Rank Aggregation with runtime and space $2^{O(\sqrt{K})} + n^{O(1)}$ for $OPT/m \leq K$. A variant uses $2^{O(\sqrt{K}\log(K))} + n^{O(1)}$ time and $n^{O(1)}$ space.*

Note that our bound in Theorem 5 is based on an upper-bound K on the scaled optimum value OPT/m, that is the average distance from input rankings to the output ranking. This is arguably a more natural parameter than OPT itself. The best previously known runtime was $n^{O(1)} + 2^{O(K)}$ [10].[1]

We also give the first fixed-parameter tractability result for our fragile ranking generalization for arity 3.

Theorem 6. *There exist parameterized subexponential algorithms for all fragile rank CSPs on tournaments with arity three (e.g. 3-*FAST *and* BetweennessTour*) with runtime $2^{O(\sqrt{K/n})} \cdot n^{O(1)}$ for $OPT \leq K$.*

For betweenness the previously best known runtime was $2^{O(K^{1/3}\log K)}$ [24]. Our result is better by a log factor in the exponent for the largest possible $K = \Theta(n^3)$ and even better for smaller K. Interestingly we can solve instances with K as large as $\Theta(n)$ in polynomial time!

We give the algorithms and the analysis of our PTAS for fragile problems in Sections 3-8 and Appendix A of this paper. We state and analyze our exact algorithms in Appendix B. We extend our results to weak fragility in Appendix C.

2 Intuition and main ideas

Our first key idea is analogous to the approximation of a differentiable function by a tangent line. Given a ranking π and any ranking CSP, the change in cost from switching to a similar ranking π' can be well approximated by the change in cost of a particular weighted feedback arc set problem (see proof of Lemma 24). Furthermore if the ranking CSP is fragile and fully dense the corresponding feedback arc set instance is a (weighted) tournament (Lemma 18). So *if* we somehow had access to a ranking similar to the optimum ranking π^* we could create this FAST instance and run the existing PTAS for FAST [20] to get a good ranking.

We do not have access to π^* but we can use a variant of the fragile techniques of [19] to get close. We pick a random sample of vertices and guess their location in the optimal ranking to within ϵn. We then create an ordering σ^1 greedily from the random sample. We show that this ordering is close to π^*, in that $|\pi^*(v) - \sigma^1(v)| = O(\epsilon n)$ for all but $O(\epsilon n)$ of the vertices (Lemma 13).

We then do a second greedy step (relative to σ^1), creating σ^2. We then identify a set U of *unambiguous* vertices for which we know $|\pi^*(v) - \sigma^2(v)| = O(\epsilon n)$ (Lemma 17). We temporarily

[1] Stated therein as runtime $2^{O(d_a)}$ where d_a is the average pairwise Kendall-Tau distance between the input rankings. We note that $d_a = \Theta(OPT/m)$ follows easily from the triangle inequality; see e.g. the classic proof that picking a random input ranking is a 2-approximation in expectation.

set aside the $O(OPT/(\epsilon n^{k-1}))$ (Lemma 16) remaining vertices. These two greedy steps are similar in spirit to previous work on ordinary (non-ranking) everywhere-dense fragile CSPs [19] but substantially more involved.

We then use σ^2 to create a FAST instance w that locally represents the CSP. Unfortunately the error in σ^2 causes the weights of w to have significant error (Lemma 20) even when $OPT \approx 0$. At first glance even an exact solution to this FAST problem would seem insufficient, for how can solving a problem similar to the desired one lead to a precisely correct solution? We show that FAST is tolerant of such errors (Lemma 24). The intuition for why this is possible is that minor adjustments to edge weights of a zero-cost FAST instance change the optimum *cost* but leave the optimum *ranking* unchanged.

Another difficulty is that the incorrect weights in FAST instance w may increase the optimum cost of w far above OPT, leaving the PTAS for FAST free to return a poor ranking. To remedy this we create a new FAST instance $\bar{w}$ by canceling weight on opposing edges, i.e. reducing w_{uv} and w_{vu} by the same amount. The resulting simplified instance $\bar{w}$ clearly has the same optimum ranking as w but a smaller optimum value. The PTAS for FAST requires that the ratio of the maximum and the minimum of $w_{uv} + w_{vu}$ must be bounded above by a constant so we limit the amount of cancellation to ensure this (Lemma 18). It turns out that this cancellation trick is sufficient to ensure that the PTAS for FAST does not introduce too much error (Lemma 21).

Finally we greedily insert the relatively few ambiguous vertices into the ranking output by the PTAS for FAST.

3 Approximation Algorithm

First we state some core notation. Throughout this paper let V refer to the set of objects (vertices) being ranked and n denotes $|V|$. Our $O(\cdot)$ hides k but not ϵ or n. Our $\tilde{O}(\cdot)$ hides $(\log(1/\epsilon))^{O(1)}$. A *ranking* is a bijective mapping from a ground set $S \subseteq V$ to $\{1, 2, 3, \ldots, |S|\}$. An *ordering* is an injection from S into $\mathbb{R}$. We use π and σ (plus superscripts) to denote orderings and rankings respectively. Let π^* denote an optimal ordering and OPT its cost. We let $\binom{n}{k}$ (for example) denote the standard binomial coefficient and $\binom{V}{k}$ denote the set of subsets of set V of size k.

For any ordering σ let $Ranking(\sigma)$ denote the ranking naturally associated with σ. To help prevent ties we relabel the vertices so that $V = \{1, 2, 3, \ldots, |V|\}$. We will often choose to place u in one of $O(1/\epsilon)$ positions $\mathcal{P}(u) = \{j\epsilon n + u/(n+1), 0 \leq j \leq 1/\epsilon\}$ (the $u/(n+1)$ term breaks ties), where $\epsilon > 0$ is the desired approximation parameter. We say that an ordering is a *bucketed ordering* if $\sigma(u) \in \mathcal{P}(u)$ for all u. Let $Round(\pi)$ denote the bucketed ordering corresponding to π (rounding down), i.e. $Round(\pi)(u)$ equals $\pi(u)$ rounded down to the nearest multiple of ϵn, plus $u/(n+1)$.

Let $v \mapsto p$ denote the ordering over $\{v\}$ which maps v to p. For set Q of vertices and ordering σ with domain including Q let $Q \mapsto \sigma$ denote the ordering over Q which maps $u \in Q$ to $\sigma(u)$, i.e. the restriction of σ to Q. For orderings σ^1 and σ^2 with disjoint domains let $\sigma^1 | \sigma^2$ denote the natural combined ordering over $Domain(\sigma^1) \cup Domain(\sigma^2)$. For example of our notations, $Q \mapsto \sigma | v \mapsto p$ denotes the ordering over $Q \cup \{v\}$ that maps v to p and $u \in Q$ to $\sigma(u)$.

A ranking k-CSP consists of a ground set V of *vertices*, an arity $k \geq 2$, and a *constraint system* c, where c is a function from rankings of k vertices to $\{0, 1\}$.[2] We say that a subset $S \subset V$ of size k is *satisfied* in ordering σ of S if $c(Ranking(\sigma)) = 0$. For brevity we henceforth abuse notation and omit the "*Ranking*" and write simply $c(\sigma)$. The objective of a ranking CSP is to find an

[2] Our results transparently generalize to the $[0, 1]$ case as well, but the 0/1 case allows simpler terminology.

ordering σ (w.l.o.g. a ranking) minimizing the number of unsatisfied constraints, which we denote by $C^c(\sigma) = \sum_{S \in \binom{Domain(\sigma)}{k}} c(S \mapsto \sigma)$. We will frequently omit the superscript c, in which case it should be understood to be the constraint system of the overall problem we are trying to solve.

Abusing notation we sometimes refer to $S \subseteq V$ as a *constraint*, when we really are referring to $c(S \mapsto \cdot)$. A constraint S is *fragile* if whenever it is satisfied making any single vertex move that changes the relative order of the vertices in S makes it unsatisfied. In other words constraint S is fragile if $c(S \rightarrow \pi) + c(S \rightarrow \pi') \geq 1$ for all rankings π and π' over S that differ by a single vertex move, i.e. $\pi' = Ranking(v \rightarrow p | S \setminus \{v\} \rightarrow \pi)$ for some $v \in S$ and $p \in (\mathbb{Z} + 1/2)$.

Our techniques handle ranking CSPs that are *fully dense* with fragile constraints, i.e. every set S of k vertices corresponds to a fragile constraint. Fully dense instances are also known as tournaments.

Let $b^c(\sigma, v, p) = \sum_{Q:\dots} c(Q \mapsto \sigma | v \mapsto p)$, where the sum is over sets $Q \subseteq Domain(\sigma) \setminus \{v\}$ of size $k-1$. Note that this definition is valid regardless of whether or not v is in $Domain(\sigma)$. The only requirement is that the range of σ excluding $\sigma(v)$ must not contain p. This ensures that the argument to $c(\cdot)$ is an ordering (injective). We will usually omit the superscript c (as with C).

We call a non-negative weight function w over the edges of the complete graph induced by some vertex set U a *FAS instance*. We can express the FAST problem in our framework by the correspondence $c(u \mapsto x | v \mapsto y) = \begin{cases} w_{vu} & \text{if } x < y \\ w_{uv} & \text{otherwise} \end{cases}$. Abusing notation slightly we also write $C^w(\sigma)$ for $C^c(\sigma)$ with the above c. More concretely $C^w(\sigma) = \sum_{u,v:\sigma(u)>\sigma(v)} w_{uv}$. Similarly we write $b^w(\sigma, v, p) = \sum_{u \neq v} \begin{cases} w_{uv} & \text{if } \sigma(u) > p \\ w_{vu} & \text{if } \sigma(u) < p \end{cases}$. Observe that FAST captures all possible fragile constraints with $k = 2$. We generalize to k-FAST as follows: a k-FAST constraint over S is satisfied by one particular ranking of S and no others.

We use the following two results from the literature.

Theorem 7 ([20]). *Let w be a FAS instance satisfying $\alpha \leq w_{uv} + w_{vu} \leq \beta$ for $\alpha, \beta > 0$ and $\beta/\alpha = O(1)$. There is a PTAS for the problem of finding a ranking π minimizing $C^w(\pi)$ with runtime $n^{O(1)} 2^{\tilde{O}(1/\epsilon^6)}$.*

Theorem 8 (e.g. [7, 21]). *For any $\delta > 0$ and constraint system c, k there is an algorithm* AddApprox *for the problem of finding a ranking π with $C(\pi) \leq C(\pi^*) + \delta n^k$. Its runtime is $n^{O(1)} 2^{\tilde{O}(1/\delta^2)}$.*

For any ordering σ with domain U let w^σ_{uv} equal the number of the constraints $\{u, v\} \subseteq S \subseteq U$ with $c(\sigma') = 1$ where (1) $\sigma' = S \setminus \{v\} \mapsto \sigma | v \mapsto p$, (2) $p = \sigma(u) - \delta$ if $\sigma(v) > \sigma(u)$ and $p = \sigma(v)$ otherwise, and (3) $\delta > 0$ is sufficiently small to put p adjacent to $\sigma(u)$. In other words if v is after u in σ it is placed immediately before v in σ'. Observe that $0 \leq w_{uv} \leq \binom{|U|-2}{k-2}$. We use the abbreviation $C^{\sigma'}(\sigma) = C^{w^{\sigma'}}(\sigma)$. The following Lemma follows easily from the definitions.

Lemma 9. *For any ordering σ we have (1) $C^\sigma(\sigma) = \binom{k}{2} C(\sigma)$ and (2) $b^{w^\sigma}(\sigma, v, \sigma(v)) = (k-1) \cdot b(\sigma, v, \sigma(v))$ for all v.*

Proof. Observe that all w_{uv} that contribute to $C^\sigma(\sigma)$ or $b^{w^\sigma}(\sigma, v, \sigma(v))$ satisfy $\sigma(u) > \sigma(v)$ and hence such w_{uv} are equal to the number of constraints containing u and v that are unsatisfied in σ. The $\binom{k}{2}$ and $k-1$ factors appear because constraints are counted multiple times. ☐

We define $\bar{w}^\sigma_{uv} = w^\sigma_{uv} - \min(\frac{1}{10}\binom{|U|-2}{k-2}, w^\sigma_{uv}, w^\sigma_{vu})$, where U is the domain of σ. Let $\bar{C}^\sigma(\sigma') = C^{\bar{w}^\sigma}(\sigma')$. Observe that w and $\bar{w}$ are equivalent from an exact solution point of view, but $\bar{w}$ has a

Algorithm 1 A $1+O(\epsilon)$-approximation for fragile rank k-CSPs in tournaments.

Input: Vertex set V, $|V|=n$, arity k, system c of fully dense arity k constraints, and approximation parameter $\epsilon > 0$.

1: Run $\textsc{AddApprox}(\epsilon^5 n^k)$ and return the result if its cost is at least $\epsilon^4 n^k$

2: Pick sets $T_1, \ldots, T_t$ uniformly at random with replacement from $\binom{V}{k-1}$, where $t = \frac{14\ln(40/\epsilon)}{\binom{k}{2}\epsilon}$. Guess (by exhaustion) bucketed ordering σ^0, which is the restriction of $Round(\pi^*)$ to the sampled vertices $\bigcup_i T_i$.

3: Compute bucketed ordering σ^1 greedily with respect to the random samples and σ^0: $\sigma^1(u) = \text{argmin}_{p\in\mathcal{P}(u)} \hat{b}(u,p)$ where $\hat{b}(u,p) = \frac{\binom{n}{k-1}}{t}\sum_{i:u\notin T_i} c(T_i \mapsto \sigma^0 \| v \mapsto p)$.

4: For each vertex v: If $b(\sigma^1, v, p) \le 13k^4\epsilon\binom{n-1}{k-1}$ for some $p \in \mathcal{P}(v)$ then call v *unambiguous* and set $\sigma^2(v)$ to the corresponding p (pick any if multiple p satisfy). Let U denote the set of unambiguous vertices, which is the domain of bucketed ordering σ^2.

5: Compute feedback arc set instance over unambiguous vertices U with weights $\bar{w}^{\sigma^2}_{uv}$ (see text). Solve it using FAST PTAS. Do single vertex moves until local optimality (with respect to FAST objective function), yielding ranking π^3 of U.

6: Create ordering σ^4 over V defined by $\sigma^4(u) = \begin{cases} \pi^3(u) & \text{if } u \in U \\ \text{argmin}_{p=v/(n+1)+j, 0\le j\le n} b(\pi^3, u, p) & \text{otherwise} \end{cases}$.
In other words insert each vertex $v \in V \setminus U$ into $\pi^3(v)$ greedily.

7: Return $\pi^4 = Ranking(\sigma^4)$.

smaller objective value for approximation purposes. In other words $C^\sigma(\pi') - C^\sigma(\pi^\circ) = \bar{C}^\sigma(\pi') - \bar{C}^\sigma(\pi^\circ)$ for all rankings π' and π°.

For any orderings σ and σ' with domain U, we say that $\{u,v\} \subseteq U$ is a σ/σ'-*inversion* if $\sigma(u)-\sigma(v)$ and $\sigma'(u)-\sigma'(v)$ have different signs. Let $d(\sigma,\sigma')$ denote the number of σ/σ'-inversions (a.k.a. Kendall Tau distance). We say that v does a *left to right* (σ, p, σ', p')-*crossing* if $\sigma(v) < p$ and $\sigma'(v) > p'$. We say that v does a *right to left* $(\sigma, p/\sigma', p')$-*crossing* if $\sigma(v) > p$ and $\sigma'(v) < p'$. We say that v does a (σ, p, σ', p')-*crossing* if v does a crossing of either sort. We say that u σ/σ'-*crosses* $p \in \mathbb{R}$ if it does a (σ, p, σ', p)-crossing.

If $OPT \ge \epsilon^4 n^k$ then the first line of the algorithm is sufficient for a PTAS so for the remainder of the analysis we assume that $OPT \le \epsilon^4 n^k$. For most of the analysis we actually need something weaker, namely that OPT is at most some sufficiently small constant times $\epsilon^2 n^k$. We only need the full $OPT \le \epsilon^4 n^k$ in one place in Section 8.

4 Runtime analysis

By Theorem 8 the additive approximation step takes time $n^{O(1)}2^{\tilde{O}(1/\epsilon^{10})}$. There are at most $(1/\epsilon)^{t\cdot(k-1)} = 2^{\tilde{O}(1/\epsilon)}$ bucketed orderings σ^0 to try. The PTAS for FAST takes time $n^{O(1)}2^{\tilde{O}(1/\epsilon^6)}$ by Theorem 7. The overall runtime is

$$n^{O(1)}2^{\tilde{O}(1/\epsilon^{10})} + 2^{\tilde{O}(1/\epsilon)} \cdot \left(n^{O(1)} + n^{O(1)}2^{\tilde{O}(1/\epsilon^6)}\right) = n^{O(1)}2^{\tilde{O}(1/\epsilon^{10})}.$$

5 Analysis of σ^1

Let $\sigma^\square = Round(\pi^*)$. Say that vertex v is *costly* if $b(\sigma^\square, v, \sigma^\square(v)) \ge 2\binom{k}{2}\epsilon\binom{n-1}{k-1}$ and *non-costly* otherwise.

The proofs of the following two Lemmas are deferred to Appendix A.

Lemma 10. *The number of costly vertices is at most* $\frac{k \cdot OPT}{\epsilon\binom{k}{2}\binom{n-1}{k-1}}$.

Lemma 11. *Let σ be an ordering of V, $v \in V$ be a vertex and $p, p' \in \mathbb{R}$. Let B be the set of vertices (excluding v) between p and p' in σ. Then $b(\sigma, v, p) + b(\sigma, v, p') \geq \frac{|B|}{k-1}\binom{n-2}{k-2}$.*

For vertex v we say that a position $p \in \mathcal{P}(v)$ is *v-out of place* if there are at least $6\binom{k}{2}\epsilon n$ vertices between p and $\sigma^{\square}(v)$ in $\sigma^{\square}$. We say vertex v is *out of place* if $\sigma^1(v)$ is v-out of place.

The proof of the following Lemma is also deferred to Appendix A.

Lemma 12. *The number of non-costly out of place vertices is at most $\epsilon n/2$ with probability at least 9/10.*

Lemma 13. *With probability at least 9/10 we have*

1. *The number of out of place vertices is at most ϵn.*
2. *The number of vertices v with $|\sigma^1(v) - \sigma^{\square}(v)| > 3k^2\epsilon n$ is at most ϵn*
3. $d(\sigma^1, \sigma^{\square}) \leq 6k^2\epsilon n^2$

Proof. By Lemma 10 and the fact $OPT \leq \epsilon^4 n^k$ we have at most $\frac{k \cdot OPT}{\binom{k}{2}\epsilon\binom{n-1}{k-1}} \leq \epsilon n/2$ costly vertices for n sufficiently large. Therefore Lemma 12 implies the first part of the Lemma.

Observe that any vertex with $|\sigma^1(v) - \sigma^{\square}(v)| > 3k^2\epsilon n \geq (6\binom{k}{2} + 1)\epsilon n$ must necessarily be v-out of place, completing the proof of the second part of the Lemma.

For the final part observe that if u and v are a $\sigma^1/\sigma^{\square}$-inversion and not among the ϵn out of place vertices then there can be at most $2 \cdot 6\binom{k}{2}\epsilon n$ vertices between $\sigma^{\square}(v)$ and $\sigma^{\square}(u)$ in $\sigma^{\square}$. Each u therefore only $24\binom{k}{2}\epsilon n$ possibilities for v. Therefore $d(\sigma^1, \sigma^{\square}) \leq \epsilon n^2 + 24\binom{k}{2}\epsilon n \cdot n/2 \leq 6\epsilon k^2 n^2$. $\square$

Our remaining analysis is deterministic, conditioned on the event of Lemma 13 holding.

6 Analysis of σ^2

The following key Lemma shows the sensitivity of $b(\sigma, v, p)$ to its first and third arguments.

Lemma 14. *For any constraint system c, k with $k \geq 2$, orderings σ and σ' over vertex set $T \subseteq V$, vertex $v \in V$ and $p, p' \in \mathbb{R}$ we have*

1. $$|b^c(\sigma, v, p) - b^c(\sigma', v, p')| \leq \binom{n-2}{k-2}(\textit{number of crossings}) + \binom{n-3}{k-3}d(\sigma, \sigma')$$
2. $$|b^c(\sigma, v, p) - b^c(\sigma', v, p')| \leq \binom{n-2}{k-2}\left(|\textit{net flow}| + k\sqrt{d(\sigma, \sigma')}\right)$$

where $\binom{n-3}{k-3} = 0$ if $k = 2$, $(\textit{net flow})$ is $|\{v \in T : \sigma'(v) > p'\}| - |\{v \in T : \sigma(v) > p\}|$, and $(\textit{number of crossings})$ is the number of $v \in T$ that do a (σ, p, σ', p')-crossing.

Proof. Fix σ, σ', T, v, p and p'. Let L (resp. R) denote the vertices in T that do left to right (resp. right to left) (σ, p, σ', p')-crossing. It is easy to see that a constraint $\{v\} \cup Q$, $Q \in \binom{T \setminus \{v\}}{k-1}$ contributes identically to $b(\sigma, v, p)$ and $b(\sigma', v, p')$ unless either:

1. Q and $(L \cup R)$ have non-empty intersection (or)

2. Q contains a σ/σ'-inversion $\{s, t\}$.

The first part of the Lemma follows easily.

Towards proving the second part we first bound $|L| + |R|$. Observe that $|L| = |R| + (\textit{net flow})$. Assume w.l.o.g. that $(\textit{net flow}) \geq 0$. Observe that every pair $v \in L$ and $w \in R$ are a σ/σ'-inversion, hence $d(\sigma, \sigma') \geq |L| \cdot |R| = (|R| + (\textit{net flow}))|R| \geq |R|^2$. We conclude that $|L| + |R| = 2|R| + (\textit{net flow}) \leq 2\sqrt{d(\sigma, \sigma')} + (\textit{net flow})$. Therefore the number of constraints of the first type is at most $\binom{n-2}{k-2}(2\sqrt{d(\sigma, \sigma')} + (\textit{net flow}))$.

To simplify we bound

$$\binom{n-3}{k-3} d(\sigma, \sigma') = \binom{n-2}{k-2} \sqrt{d(\sigma, \sigma')} \cdot \frac{k-2}{n-2} \cdot \sqrt{d(\sigma, \sigma')}$$
$$\leq \binom{n-2}{k-2} \sqrt{d(\sigma, \sigma')} \cdot (k-2) \frac{\sqrt{n(n-1)/2}}{n-2} \leq (k-2) \binom{n-2}{k-2} \sqrt{d(\sigma, \sigma')}$$

for sufficiently large n. □

We define the *net σ/σ'-flow across p* to be $|\{\, v \in T : \sigma'(v) > p \,\}| - |\{\, v \in T : \sigma(v) > p \,\}|$ where T is the domain of σ and σ'. Note that this is a specialization of the *net flow* in Lemma 14 to the case $p = p'$. Usefully such a flow is zero if σ and σ' are both *rankings*.

Corollary 15. *Let π and π' be rankings over vertex set U and w a FAST instance over U. Then $|b^w(\pi, v, p) - b^w(\pi', v, p)| \leq 2(\max_{r,s} w_{rs})\sqrt{d(\pi, \pi')}$ for all v and $p \in \mathbb{R} \setminus \mathbb{Z}$.*

Lemma 16. $|V \setminus U| \leq \frac{k \cdot OPT}{\epsilon \binom{k}{2} \binom{n-1}{k-1}} = O(\frac{n}{\epsilon} \cdot \frac{OPT}{n^k})$.

Proof. Observe that the number of vertices that $\sigma^{\Box}/\sigma^1$-cross a particular p is at most $2 \cdot 6k^2 \epsilon n$ by Lemma 13 (first part). Therefore we apply Lemmas 13 and 14, yielding

$$|b(\sigma^{\Box}, v, p) - b(\sigma^1, v, p)| \leq \binom{n-2}{k-2} 12k^2 \epsilon n + \binom{n-3}{k-3} 6k^2 \epsilon n^2 \leq 12 \epsilon k^4 \binom{n-1}{k-1} \tag{1}$$

for all v and p.

Fix a non-costly v. By definition of costly $b(\sigma^{\Box}, v, \sigma^{\Box}(v)) \leq 2\binom{k}{2}\epsilon\binom{n-1}{k-1} \leq k^4 \epsilon \binom{n-1}{k-1}$, hence $b(\sigma^1, v, \sigma^{\Box}(v)) \leq 13k^4 \epsilon \binom{n-1}{k-1}$, so $v \in U$.

Finally recall Lemma 10. □

We define $\pi^{\circledast}$ to be the ranking induced by the restriction of π^* to U.

Lemma 17. *All vertices in the unambiguous set U satisfy $|\sigma^2(v) - \pi^{\circledast}(v)| = O(\epsilon n)$.*

Proof. Since π^* is a ranking the number of vertices $|B|$ between $\pi^*(v)$ and $\sigma^2(v)$ in π^* is at least $|\pi^*(v) - \sigma^2(v)| - 1$. Therefore by Lemma 11 we have

$$\frac{|\pi^*(v) - \sigma^2(v)| - 1}{n-1} \binom{n-1}{k-1} \leq b(\pi^*, v, \sigma^2(v)) + b(\pi^*, v, \pi^*(v)) \qquad \text{(Lemma 11)}$$
$$\leq 2b(\pi^*, v, \sigma^2(v)) \qquad \text{(Optimality of } \pi^*\text{).}$$

We proceed

$$\begin{aligned} b(\pi^*, v, \sigma^2(v)) &\leq b(\sigma^{\square}, v, \sigma^2(v)) + O(\epsilon n^{k-1}) && \text{(Lemma 14, part one)} \\ &\leq b(\sigma^1, v, \sigma^2(v)) + O(\epsilon n^{k-1}) + O(\epsilon n^{k-1}) && (1) \\ &= O(\epsilon n^{k-1}) && \text{(Definition of } U) \end{aligned}$$

hence we conclude $|\pi^*(v) - \sigma^2(v)| = O(\epsilon n)$.

Finally by conclude

$$\begin{aligned} |\pi^{\circledast}(v) - \sigma^2(v)| &\leq |\pi^{\circledast}(v) - \pi^*(v)| + |\pi^*(v) - \sigma^2(v)| = |\pi^{\circledast}(v) - \pi^*(v)| + O(\epsilon n) \\ &\leq \frac{k \cdot OPT}{\epsilon\binom{k}{2}\binom{n-1}{k-1}} + O(\epsilon n) && \text{(Lemma 16)} \\ &= O(\epsilon n). \end{aligned}$$

□

7 Analysis of π^3

Note that all orderings and costs in this section are over U, not V. We note that $|U| \approx |V|$ so $|U| = \Theta(n)$ by Lemma 16.

Lemma 18. $(1 - 2/10)\binom{|U|-2}{k-2} \leq \bar{w}^{\sigma^2}_{uv} + \bar{w}^{\sigma^2}_{vu} \leq (2 - 2/10)\binom{|U|-2}{k-2}$, *i.e.* $\bar{w}^{\sigma^2}$ *is a weighted FAST instance.*

Proof. By fragility each of the $\binom{n-2}{k-2}$ constraints $S \supseteq \{u, v\}$ contributes between 1 and 2 to $w_{uv}+w_{vu}$. The Lemma follows from the definition of $\bar{w}$. □

We define the shorthand $OPT_U = C(\pi^{\circledast})$.

Lemma 19. *Assume ranking* π *and ordering* σ *satisfy* $|\pi(u) - \sigma(u)| = O(\epsilon n)$ *for all* u*. For any* u, v*, let* N_{uv} *denote the number of* $S \supset \{u, v\}$ *such that not all pairs* $\{s, t\} \neq \{u, v\}$ *are in the same order in* σ *and* π*. We have* $N_{uv} = O(\epsilon n^{k-2})$.

Proof. Such a pair $\{s, t\}$ must satisfy $|\pi(s) - \pi(t)| = 2 \cdot O(\epsilon n)$, but few constraints contain such a pair. □

Lemma 20. *The following inequalities hold:*

1. $w^{\sigma^2}_{uv} \leq w^{\pi^{\circledast}}_{uv} + O(\epsilon n^{k-2})$
2. $\bar{w}^{\sigma^2}_{uv} \leq (1 + O(\epsilon))w^{\pi^{\circledast}}_{uv}$

Proof. The only constraints $S \supset \{u, v\}$ that contribute differently to the left- and right-hand sides of the first part are those containing a $\{s, t\} \neq \{u, v\}$ that are a $\sigma^2/\pi^{\circledast}$-inversion. By Lemmas 17 and 19 we can bound the number of such constraints by $O(\epsilon n^k)$, completing the proof of the first part.

If $w^{\pi^{\circledast}}_{uv} \geq (1/2)\binom{|U|-2}{k-2}$ the claim follows from the first part and the trivial fact $\bar{w} \leq w$. Otherwise by the first part we have $w^{\sigma^2}_{uv} < 0.6\binom{|U|-2}{k-2}$. Therefore by Lemma 18 $w^{\sigma^2}_{vu} > 0.2\binom{|U|-2}{k-2}$ hence $\bar{w}^{\sigma^2}_{uv} = w^{\sigma^2}_{uv} - \min(0.1\binom{|U|-2}{k-2}, w^{\sigma^2}_{uv}) = \min(w^{\sigma^2}_{uv} - 0.1\binom{|U|-2}{k-2}, 0) \leq \min(w^{\pi^{\circledast}}_{uv}, 0) \leq w^{\pi^{\circledast}}_{uv}$ using the first part of the Lemma in the penultimate inequality.

□

Lemma 21.

1. $\bar{C}^{\sigma^2}(\pi^{\circledast}) \le (1+O(\epsilon))\binom{k}{2} OPT_U$
2. $\bar{C}^{\sigma^2}(\pi^3) \le (1+O(\epsilon))\binom{k}{2} OPT_U$
3. $\bar{C}^{\sigma^2}(\pi^3) - \bar{C}^{\sigma^2}(\pi^{\circledast}) = O(\epsilon OPT_U)$

Proof. From the second part of Lemma 20 and Lemma 9 we conclude that

$$\bar{C}^{\sigma^2}(\pi^{\circledast}) \le (1+O(\epsilon))C^{\pi^{\circledast}}(\pi^{\circledast}) = (1+O(\epsilon))\binom{k}{2}OPT_U.$$

proving the first part of this Lemma.

The PTAS for FAST guarantees

$$\bar{C}^{\sigma^2}(\pi^3) \le (1+O(\epsilon))\bar{C}^{\sigma^2}(\pi^{\circledast}), \tag{2}$$

which combined with the first part of this Lemma yields the second part.

Finally the first part of Lemma 20 followed by the first part of this Lemma imply

$$\bar{C}^{\sigma^2}(\pi^3) - \bar{C}^{\sigma^2}(\pi^{\circledast}) \le O(\epsilon)C^{\sigma^2}(\pi^{\circledast}) \le O(\epsilon OPT_U),$$

completing the proof of the third part of this Lemma. □

Lemma 22. $d(\pi^3, \pi^{\circledast}) = O(OPT_U/n^{k-2})$

Proof. π^3 and $\pi^{\circledast}$ both have cost at most $2OPT_U$ (Lemma 21, first and second parts) for the FAST instance $\bar{w}^{\sigma^2}$ (Lemma 18). □

Lemma 23. *We have $|\pi^3(v) - \pi^{\circledast}(v)| = O(\epsilon n)$ for all $v \in U$.*

Proof. In this proof we write w (resp. $\bar{w}$) as a short-hand for w^{σ^2} (resp. $\bar{w}^{\sigma^2}$). By Lemma 18 and local optimality of π^3 we have

$$\begin{aligned}(|\pi^3(v) - \pi^{\circledast}(v)| - 1)(1-2/10)\binom{|U|-2}{k-2} &\le b^{\bar{w}}(\pi^3, v, \pi^{\circledast}(v)+1/2) + b^{\bar{w}}(\pi^3, v, \pi^3(v)) \\ &\le 2b^{\bar{w}}(\pi^3, v, \pi^{\circledast}(v)+1/2).\end{aligned}$$

Now apply Corollary 15

$$b^{\bar{w}}(\pi^3, v, \pi^{\circledast}(v)+1/2) \le b^{\bar{w}}(\pi^{\circledast}, v, \pi^{\circledast}(v)) + 2\sqrt{d(\pi^{\circledast}, \pi^3)}(2-2/10)\binom{|U|-2}{k-2}$$

and then recall $\sqrt{d(\pi^{\circledast}, \pi^3)} = O(\epsilon n)$ by Lemma 22 and the assumption that OPT is small.

Next

$$\begin{aligned}b^{\bar{w}}(\pi^{\circledast}, v, \pi^{\circledast}(v)) &\le (1+O(\epsilon))b^{w^{\pi^{\circledast}}}(\pi^{\circledast}, v, \pi^{\circledast}(v)) && \text{(Second part of Lemma 20)} \\ &= (1+O(\epsilon))b(\pi^{\circledast}, v, \pi^{\circledast}(v)) && \text{(Lemma 9)}\end{aligned} \tag{3}$$

Finally

$$\begin{aligned}b(\pi^{\circledast}, v, \pi^{\circledast}(v)) &\le b(\sigma^1, v, \sigma^2(v)) + O(n^{k-2}(\epsilon n + \sqrt{\epsilon^2 n^2})) && \text{(Lemmas 14, 13 and 17)} \\ &= O(\epsilon n^{k-1}) && (v \in U).\end{aligned}$$

which completes the proof of the Lemma. □

Lemma 24. $C(\pi^3) \leq (1 + O(\epsilon))OPT_U$.

Proof. First we claim that

$$|(C(\pi^3) - C(\pi^{\circledast})) - (C^{\sigma^2}(\pi^3) - C^{\sigma^2}(\pi^{\circledast}))| \leq E_1, \tag{4}$$

where E_1 is the number of constraints that contain one pair of vertices u, v in different order in π^3 and $\pi^{\circledast}$ and another pair $\{s, t\} \neq \{u, v\}$ with relative order in π^3, $\pi^{\circledast}$ and σ^2 not all equal. Indeed constraints ordered identically in π^3 and $\pi^{\circledast}$ contribute zero to both sides of (4), regardless of σ^2. Consider some constraint S containing a $\pi^3(v)/\pi^{\circledast}$-inversion $\{u, v\} \subset S$. If the restrictions of the three orderings to S are identical except possibly for swapping u, v then S contributes equally to both sides of (4), proving the claim.

To bound E_1 observe that the number of inversions u, v is $d(\pi^3, \pi^{\circledast}) \equiv D$. For any u, v Lemmas 23, 17 and 19 allow us to show at most $O(\epsilon n^{k-2})$ constraints contribute, so $E_1 = O(D\epsilon n^{k-2}) = O(\epsilon OPT_U)$ (Lemma 22).

Finally bound $C^{\sigma^2}(\pi^3) - C^{\sigma^2}(\pi^{\circledast}) = \bar{C}^{\sigma^2}(\pi^3) - \bar{C}^{\sigma^2}(\pi^{\circledast}) = O(\epsilon OPT_U)$, where the equality follows from the definition of w and the inequality is the third part of Lemma 21. □

8 Analysis of π^4

We now prove Theorem 2, that is

$$C(\pi^4) \leq (1 + O(\epsilon))OPT. \tag{5}$$

We consider three contributions to these costs separately: constraints with 0, 1, or 2+ vertices in $V \setminus U$.

The contribution of constraints with 0 vertices in $V \setminus U$ to the left- and right-hand sides of (5) are clearly $C(\pi^3)$ and $C(\pi^{\circledast})$ respectively. We showed $C(\pi^3) \leq C(\pi^{\circledast}) + O(\epsilon)OPT_U$ in Lemma 24.

Second we consider the contribution of constraints with exactly 1 vertex in $V \setminus U$. Consider some $v \in V \setminus U$. We want to compare $b(\pi^3, v, \sigma^4(v))$ and $b((U \mapsto \pi^*), v, \pi^*(v))$. Let p be the half-integer so that $Ranking(v \mapsto p | U \mapsto \pi^{\circledast}) = Ranking(v \mapsto \pi^*(v) | U \mapsto \pi^*)$. The algorithm's greedy choice minimizes $b(\pi^3, v, \sigma^4(v))$ so $b(\pi^3, v, \sigma^4(v)) \leq b(\pi^3, v, p)$. Now using Lemmas 14 and 22 we have $b(\pi^3, v, p) \leq b(\pi^{\circledast}, v, p) + O(\sqrt{d(\pi^3, \pi^{\circledast})} n^{k-2}) = b(\pi^{\circledast}, v, p) + O(\sqrt{OPT/n^k} n^{k-1}$. Note $b(\pi^{\circledast}, v, p) = b((U \mapsto \pi^*), v, \pi^*(v))$. Let $\gamma = OPT/n^k$. We conclude by Lemma 16 that the contribution of constraints with exactly 1 vertex in $V \setminus U$ is $O(|V \setminus U| \sqrt{OPT/n^k} n^{k-1}) = O(\frac{\gamma^{3/2} n^k}{\epsilon}) = O(\epsilon OPT)$.

Finally by Lemma 16 there are at most $|V \setminus U|^2 n^{k-2} = O((\frac{\gamma}{\epsilon})^2 n^2 n^{k-2}) = O(\epsilon^2 OPT)$ constraints containing two or more vertices from $V \setminus U$.

This ends the analysis of our algorithm.

Acknowledgements

We would like to thank Venkat Guruswami, Claire Mathieu, Prasad Raghavendra and Alex Samorodnitsky for interesting remarks and discussions.

References

[1] N. Ailon. Aggregation of Partial Rankings, p-ratings and top-m Lists. In *18th SODA*, pages 415–424, 2007.

[2] N. Ailon. Aggregation of partial rankings, p-ratings and top-m lists. In *Procs. 18th ACM-SIAM SODA*, pages 415–424, 2007. Journal version to appear in *Algorithmica*.

[3] N. Ailon and N. Alon. Hardness of Fully Dense Problems. *Inf. Comput.*, 205:1117–1129, 2007.

[4] N. Ailon, M. Charikar, and A. Newman. Aggregating Inconsistent Information: Ranking and Clustering. *J. ACM*, 55, 2008.

[5] N. Alon, W. Fernandez de la Vega, R. Kannan, and M. Karpinski. Random Sampling and Approximation of MAX-CSP Problems. In *34th ACM STOC*, pages 232–239, 2002. Journal version in J. Comput. System Sciences 67 (2003), pp. 212-243.

[6] N. Alon, D. Lokshtanov, and S. Saurabh. Fast FAST. In *36th ICALP*, pages 49–58, 2009.

[7] S. Arora, D. Karger, and M. Karpinski. Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems. In *27th ACM STOC*, pages 284–293, 1995. Journal version in J. Comput. System Sciences 58 (1999), pp. 193-210.

[8] C. Bazgan, W. Fernandez de la Vega, and M. Karpinski. Polynomial Time Approximation Schemes for Dense Instances of the Minimum Constraint Satisfaction Problem. *Random Structures and Algorithms*, 23:73–91, 2003.

[9] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny scores. In *AAIM '08: Procs. 4th int'l conf. on Algorithmic Aspects in Information and Management*, pages 60–71, 2008.

[10] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. How similarity helps to efficiently compute Kemeny rankings. In *AAMAS '09: 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 657–664, 2009. Journal version in Theoretical Computer Science 410 (2009) pp. 4554–4570.

[11] M. Braverman and E. Mossel. Noisy sorting without resampling. In *Procs. 19th ACM-SIAM SODA*, pages 268–276, 2008.

[12] M. Charikar, V. Guruswami, and R. Manokaran. Every Permutation CSP of Arity 3 is Approximation Restitant. In *24th IEEE CCC*, 2009.

[13] B. Chor and M. Sudan. A Geometric Approach to Betweenness. *SIAM J. Discrete Math.*, 11:511–523, 1998.

[14] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by Weighted Number of Wins gives a Good Ranking for Weighted Tournaments. In *Procs. 17th ACM-SIAM SODA*, pages 776–782, 2006.

[15] M. Dom, J. Guo, F. Hüffner, R. Niedermeier, and A. Truß. Fixed-Parameter Tractability Results for Feedback Set Problems in Tournaments. In *LNCS*, volume 3998, pages 320–331. Springer, 2006.

[16] W. Fernandez de la Vega, R. Kannan, and M. Karpinski. Approximation of Global MAX–CSP Problems. Technical Report TR06-124, Electronic Colloquim on Computation Complexity, 2006.

[17] A. Frieze and R. Kannan. Quick Approximation to Matrices and Applications. *Combinatorica*, 19:175–220, 1999.

[18] G. Gutin, E. J. Kim, M. Mnich, and A. Yeo. Ordinal embedding relaxations parameterized above tight lower bound. *CoRR/arXiv*, abs/0907.5427, 2009.

[19] M. Karpinski and W. Schudy. Linear Time Approximation Schemes for the Gale-Berlekamp Game and Related Minimization Problems. In *41st ACM STOC*, pages 313–322, 2009.

[20] C. Mathieu and W. Schudy. How to Rank with Few Errors. In *39th ACM STOC*, pages 95–103, 2007. In Submission `http://www.cs.brown.edu/~ws/papers/fast_journal.pdf`, 2009.

[21] C. Mathieu and W. Schudy. Yet Another Algorithm for Dense Max Cut: Go Greedy. In *Proc. 19th ACM-SIAM SODA*, pages 176–182, 2008.

[22] J. Opatrny. Total Ordering Problem. *SIAM J. Comput.*, 8:111–114, 1979.

[23] M. Rudelson and R. Vershynin. Sampling from Large Matrices: An Approach through Geometric Functional Analysis. *J. ACM*, 54:21, 2007.

[24] S. Saurabh. Chromatic coding and universal (hyper-) graph coloring families. In *Parameterized Complexity News*, pages 49–58, June 2009. `http://mrfellows.net/Newsletters/2009June_FPT_News.pdf`.

Appendix A Proof of Lemmas 10 and 11

Proof of Lemma 10. First observe that for any costly v we have

$$2\binom{k}{2}\epsilon\binom{n-1}{k-1} \le b(\sigma^{\Box}, v, \sigma^{\Box}(v)) \le b(\pi^*, v, \pi^*(v)) + \epsilon\binom{k}{2}\cdot\binom{n-1}{k-1}$$

since only at most a $\epsilon\binom{k}{2}$ fraction of the $\binom{n-1}{k-1}$ possible constraints contain a $\pi^*/\sigma^{\Box}$-inversion. Therefore

$$b(\pi^*, v, \pi^*(v)) \ge 2\binom{k}{2}\epsilon\binom{n-1}{k-1} - \epsilon\binom{k}{2}\cdot\binom{n-1}{k-1} = \epsilon\binom{k}{2}\cdot\binom{n-1}{k-1}$$

Secondly observe that $kC(\pi^*) = \sum_v b(\pi^*, v, \pi^*(v)) \ge (\text{number costly})\epsilon\binom{k}{2}\binom{n-1}{k-1}$, completing the proof. $\Box$

Proof of Lemma 11. By definition

$$b(\sigma, v, p) + b(\sigma, v, p') = \sum_{Q:\cdots}\left[c(Q \mapsto \sigma | v \mapsto p) + c(Q \mapsto \sigma | v \mapsto p')\right] \tag{6}$$

where the sum is over sets $Q \subseteq U \setminus \{v\}$ of $k-1$ vertices. Observe that the quantity in brackets in (6) is at least 1 for every Q that contains a vertex between p and p' by fragility. Finally note the number of such Q is at least $\frac{|B|}{k-1}\binom{n-2}{k-2}$, proving the Lemma. $\Box$

Proof of Lemma 12. Focus on some $v \in V$ and $p \in \mathcal{P}(v)$. From the definition of out-of-place and Lemma 11 we have

$$b(\sigma^{\Box}, v, \sigma^{\Box}) + b(\sigma^{\Box}, v, p) \ge \frac{6\binom{k}{2}\epsilon n}{k-1}\binom{n-2}{k-2} \ge 6\epsilon\binom{k}{2}\binom{n-1}{k-1}$$

for any v-out of place p. Next recall that for costly v we have

$$b(\sigma^{\Box}, v, \sigma^{\Box}(v)) \le 2\binom{k}{2}\epsilon\binom{n-1}{k-1} \tag{7}$$

hence

$$b(\sigma^{\Box}, v, p) \ge 4\binom{k}{2}\epsilon\binom{n-1}{k-1} \tag{8}$$

for any v-out of place p.

Recall that

$$\hat{b}(v, p) = \frac{\binom{n}{k-1}}{t}\sum_{i:v\notin T_i} c(T_i \mapsto \sigma^0 | v \mapsto p)$$

for any p. Each term of the sum is a $0/1$ random variable with mean $\mu(p) = \frac{1}{\binom{n}{k-1}}\sum_{Q\in\binom{V}{k-1}:v\notin Q} c(Q \mapsto \sigma^{\Box} | v \mapsto p) = \frac{1}{\binom{n}{k-1}} b(\sigma^{\Box}, v, p)$. Therefore $\mathbf{E}\left[\hat{b}(v,p)\right] = b(\sigma^{\Box}, v, p)$. We can bound $\mu(\sigma^{\Box}(v)) \le 2\binom{k}{2}\epsilon\binom{n-1}{k-1}/\binom{n}{k-1} \equiv M$ using (7). For any v-out of place p we can bound $\mu(p) \ge 2M$ by (8).

We can bound the probability that sum in $\hat{b}(v, \sigma^{\Box}(v))$ is at least $(1+1/3)Mt$ using a Chernoff bound as

$$\exp(-(1/3)^2 Mt/3) \le \exp\left(-\frac{1}{9}\cdot\frac{1}{\binom{n}{k-1}}\cdot 2\binom{k}{2}\epsilon\binom{n-1}{k-1}\cdot\frac{14\ln(40/\epsilon)}{\binom{k}{2}\epsilon}\cdot\frac{1}{3}\right) \le \epsilon/40$$

for sufficiently large n. Similarly for any v-out of place p we can bound the probability that $\hat{b}(v,p)$ is at most $(1-1/3)Mt$ by $\exp(-(1/3)^2 Mt/2) \leq (\epsilon/40)^3$. Therefore by union bound the probability of some v-out of place p having $\hat{b}(v,p)$ too small is at most $\epsilon^2/40^3 \leq \epsilon/40$. Clearly $4(1-1/3) \geq 2(1+1/3)$ so each vertex v is out of place with probability at least $\epsilon/20$. A Markov bound completes the proof. □

Appendix B Exact algorithms

Our exact algorithms are based on a few simple ideas. We describe our techniques for exact FAST here and defer discussion of the other problems until later. Firstly any two low-cost rankings for a FAST problem are nearby in Kendall-Tau distance. Secondly two rankings that are Kendall-Tau distance D apart are equivalent to within additive $O(\sqrt{D})$ in how good each location for each a vertex is (Corollary 15). Thirdly a consequence of fragility is that most vertices (in a low-cost instance) have a vee-shaped cost versus position curve (Lemma 26), and optimal rankings are locally optimal so we know that each vertex belongs at the bottom of its curve. The uncertainty in this curve by $\sqrt{D}$ causes an uncertainty in the optimal position also around $\sqrt{D}$ (Lemma 25). Our algorithm simply computes uncertainties $r(v)$ in the positions of all of the vertices v and solves a dynamic program for the optimal ranking that is near a particular constant-factor approximate ranking. We remark that Braverman and Mossel [11] and Betzler et al. [9, 10] previously applied dynamic programming to FAST and KRA.

The kernelization algorithm of Dom et al. [15] allows an arbitrary FAST instance of cost $OPT \leq K$ to be reduced to one with $O(K^2)$ in time $n^{O(1)}$. To achieve the results of Theorem 4 we use a constant factor approximation algorithm to upper-bound OPT, run this kernelization algorithm, and then run our general algorithm.

Lemma 25. *In Algorithm 2 we have $|\pi^*(v) - \pi^4(v)| \leq r(v)$ for all $v \in V$ where π^* is an optimal ranking of V.*

Proof. We have a tournament so $d(\pi^*, \pi^4) \leq C(\pi^*) + C(\pi^4) \leq 2C(\pi^4)$. By Lemma 15 therefore

$$|b(\pi^*, v, j+1/2) - b(\pi^4, v, j+1/2)| \leq 8\sqrt{2C(\pi^4)} \tag{9}$$

for any $j \in \mathbb{Z}$.

Algorithm 2 Exact algorithm for FAST and KRA. If dynamic programming is used in the last line the runtime and space are both $n^{O(1)} + 2^{O(\sqrt{OPT})}$. If divide-and-conquer is used the runtime is $n^{O(1)} + 2^{O(\sqrt{OPT}\log OPT)}$ and the space is $n^{O(1)}$.

Input: Vertex set V_0, constraint system c_0.

1: Compute a kernel with vertex set V, $|V| = O(OPT^2)$, and constraint system c (used for rest of algorithm) [15, 10]. Hereafter interpret notations such as $C(\cdot)$ and n relative to instance V, c, not V_0, c_0.
2: Sort the kernel by wins [14], yielding ranking π^4 of V.
3: Set $r(v) = 16\sqrt{2C(\pi^4)} + 2b(\pi^4, v, \pi^4(v))$ for all $v \in V$.
4: Use dynamic programming or divide-and-conquer (Details: Lemma 27) to find the optimal ranking π^5 with $|\pi^5(v) - \pi^4(v)| \leq r(v)$ for all v.
5: "Undo" the kernel step, extending ranking π^5 of the kernel into a ranking of V_0 as described in [15, 10].

Algorithm 3 Exact algorithm for fragile ranking 3-CSPs in tournaments. The runtime is $n^{O(1)}2^{O(\sqrt{OPT/n})}$.

Input: Vertex set V

Use our PTAS to construct a two-approximate ranking π^{good}.
for Each π^4 considered by our PTAS when constructing a 2-approximation **do**
  if $C(\pi^4) \le 2C(\pi^{good})$ **then**
    Set $r(v) = \alpha_1\sqrt{C(\pi^4)/n} + \alpha_2 b(\pi^4, v, \pi^4(v))/n$, where α_1 and α_2 are absolute constants.
    Use dynamic programming (See Lemma 27) to find the optimal ranking π^5 with $|\pi^5(v) - \pi^4(v)| \le r(v)$ for all v.
  end if
end for
Return the best of the π^5 rankings.

Fix $v\ inV$. We conclude

$$\begin{aligned}
|\pi^*(v) - \pi^4(v)| &\le b(\pi^4, v, \pi^4(v)) + b(\pi^4, v, \pi^*(v)) && \text{(Fragility)} \\
&= b(\pi^4, v, \pi^*(v) + 1/2) + b(\pi^4, v, \pi^4(v) + 1/2) && (\pi^4 \text{ is a ranking}) \\
&\le b(\pi^*, v, \pi^*(v) + 1/2) + 8\sqrt{2C(\pi^4)} + b(\pi^4, v, \pi^4(v) + 1/2) && (9) \\
&\le b(\pi^*, v, \pi^4(v) + 1/2) + 8\sqrt{2C(\pi^4)} + b(\pi^4, v, \pi^4(v) + 1/2) && (\text{Optimality of } \pi^*) \\
&\le 16\sqrt{2C(\pi^4)} + 2b(\pi^4, v, \pi^4(v) + 1/2) && (9) \\
&= r(v) && (\text{Definition of } r(v)).
\end{aligned}$$

□

Lemma 26. *In Algorithm 2 we have* $\max_j |\{ v \in V : |\pi^4(v) - j| \le r(v) \}| = O(\sqrt{OPT})$.

Proof. Fix j. Let $R = \{ v \in V : |\pi^4(v) - j| \le r(v) \}$, whose cardinality we are trying to bound. We say $v \in V$ is *pricey* if $b(\pi^4, v, \pi^4(v)) > \sqrt{2C(\pi^4)}$. Clearly (see also proof of Lemma 10) $2C(\pi^4) = \sum_v b(\pi^4, v, \pi^4(v)) \ge$ (number pricey)$\sqrt{2C(\pi^4)}$ hence the number of pricey vertices is at most $2C(\pi^4)/(\sqrt{2C(\pi^4)} = \sqrt{2C(\pi^4)}$. All non-pricey vertices in R have $|\pi^4(v) - j| \le 2 \cdot \sqrt{2C(\pi^4)}$, so at most $2\sqrt{2C(\pi^4)} + 1$ non-pricey vertices are in R. We conclude $|R| \le 3\sqrt{2C(\pi^4)} + 1 = O(\sqrt{OPT})$ since π^4 is a 5-approximation [14]. □

Lemma 27. *For $k \in \{2, 3\}$ there is a dynamic program that finds the optimal ranking π^5 with $|\pi^5(v) - \pi^4(v)| \le r(v)$ for all v, with space and runtime $O(|V|^k)2^\psi$ where $\psi = \max_j |\{ v \in V : |\pi^4(v) - j| \le r(v) \}|$. A divide and conquer variant has runtime $O(|V|^k)2^{O(\psi \log |V|)}$ and $|V|^{O(1)}$ space.*

Proof. Say that a set $S \subseteq V$ is *valid* if it contains all vertices v with $\pi^4(v) \le |S| - r(v)$ and no vertex v with $\pi^4(v) \ge |S| + r(v)$. Observe that for any s the valid sets of size s are uncertain about at most ψ vertices, hence there are at most $n2^\psi$ valid sets.

We say that a ranking π of valid set S is *valid* if $\{ v : \pi(v) \le j \}$ is a valid set for all $0 \le j \le |S|$. It is easy to see that a ranking π is valid if and only if satisfies $|\pi(v) - \pi^4(v)| \le r(v)$ for all v.

For any ranking π over S let $C'(\pi)$ denote the part of the cost shared by all orderings with prefix π. That is, the cost of all constraints with at most 1 vertex outside S.[3]

[3] For $k = 2$ (FAST) it would be more natural to use $C(\pi)$ instead, but this works better for $k = 3$.

One can easily see the following optimal substructure property: prefixes of an optimal (w.r.t. C') valid ranking are optimal (w.r.t. C') valid rankings themselves.

For any valid set S let $\kappa(S)$ denote the C' cost of the optimal (w.r.t. C') valid ranking of S. The dynamic program for FAST is

$$\kappa(S) = \min_{v \in S : S \setminus \{v\} \text{ is valid}} \left[C(S \setminus \{v\}) + \sum_{q \in V \setminus S} c(v \mapsto 2 | q \mapsto 3) \right].$$

and for betweenness

$$\kappa(S) = \min_{v \in S : S \setminus \{v\} \text{ is valid}} \left[C(S \setminus \{v\}) + \sum_{u \in S \setminus \{v\}} \sum_{q \in V \setminus S} c(u \mapsto 1 | v \mapsto 2 | q \mapsto 3) \right].$$

The space-efficient variant evaluates κ using divide and conquer instead of dynamic programming, similar to [15]. Details deferred. $\square$

Proof of Theorems 4 and 5. Algorithm 2 is correct by Lemma 25. Lemmas 26 and 27 allow us to bound the runtime and space requirements of the dynamic program. $\square$

Lemma 28. *During the iteration of Algorithm 3 that guesses σ^0 correctly we have $|\pi^*(v) - \pi^4(v)| \leq r(v)$ for all $v \in V$ where π^* is an optimal ranking of V.*

Proof. By Lemma 22 we have $d(\pi^{\circledast}, \pi^3) = O(OPT/n^{3-2})$. This together with Lemma 16 imply that

$$d(\pi^*, \pi^4) = O(OPT/n^{3-2} + n \cdot OPT/(\epsilon n^{3-1})) = O(OPT/(\epsilon n))$$

By Lemma 14 therefore

$$|b(\pi^*, v, j + 1/2) - b(\pi^4, v, j + 1/2)| = O(n\sqrt{OPT/(\epsilon n)}) \tag{10}$$

for any $j \in \mathbb{Z}$.

Fix $v \in V$. We conclude

$$\begin{aligned}
|\pi^*(v) - \pi^4(v)| \binom{n}{1} &\leq b(\pi^4, v, \pi^4(v) + 1/2) + b(\pi^4, v, \pi^*(v) + 1/2) && \text{(Lemma 11)} \\
&\leq b(\pi^*, v, \pi^*(v) + 1/2) + O(\sqrt{nC(\pi^4)/\epsilon}) + b(\pi^4, v, \pi^4(v) + 1/2) && (10) \\
&\leq b(\pi^*, v, \pi^4(v) + 1/2) + O(\sqrt{nC(\pi^4)/\epsilon}) + b(\pi^4, v, \pi^4(v) + 1/2) && \text{(Optimality of } \pi^*) \\
&\leq O(\sqrt{nC(\pi^4)/\epsilon}) + 2b(\pi^4, v, \pi^4(v) + 1/2) && (9) \\
&= r(v)n && \text{(Definition of } r(v)).
\end{aligned}$$

$\square$

Lemma 29. *We have* $\max_j |\{ v \in V \ : \ |\pi^4(v) - j| \leq r(v) \}| = O(\sqrt{C(\pi^4)/n})$.

Proof. We proceed analogously to the proof of Lemma 26. Fix j. Let $R = \{ v \in V : |\pi^4(v) - j| \leq r(v) \}$, whose cardinality we are trying to bound. We say $v \in V$ is *pricey* if $b(\pi^4, v, \pi^4(v))/n > \sqrt{2C(\pi^4)/n}$. Clearly (see also proof of Lemma 10) $3C(\pi^4) = \sum_v b(\pi^4, v, \pi^4(v)) \geq$ (number pricey)$n\sqrt{2C(\pi^4)/n}$ hence the number of pricey vertices is at most $3C(\pi^4)/(\sqrt{2nC(\pi^4)}) = \sqrt{2C(\pi^4)/n}$. All non-pricey vertices in R have $|\pi^4(v) - j| \leq 2 \cdot \sqrt{2C(\pi^4)/n}$, so at most $2\sqrt{2C(\pi^4)/n} + 1$ non-pricey vertices are in R. We conclude $|R| \leq 3\sqrt{2C(\pi^4)/n} + 1 = O(\sqrt{C(\pi^4)/n})$. $\square$

Proof of Theorem 6. Lemmas 27 and 29, plus the test of the "if", allow us to bound the runtime and space requirements of the dynamic program used by Algorithm 3by $n^{O(1)}2^{O(\sqrt{C(\pi^{good})/n})}$, which is of the correct order since $\pi^{good} \leq 2\pi^*$. The for loop is over a constant number of options and is therefore irrelevant.

For correctness we focus on the iteration of Algorithm 3 that guesses σ^0 correctly. The approximation guarantee of our PTAS holds for this iteration so we have $\pi^4 \leq 2\pi^* \leq 2\pi^{good}$ and hence the "if" is passed. By Lemma 25 π^* is among the orders the dynamic program considers. $\square$

Appendix C Weak Fragility

A constraint S is *weakly fragile* if $c(S \to \pi) + c(S \to \pi') \geq 1$ for all rankings π and π' that differ by a swap of the first two vertices, the last two, or cyclic shift of a single vertex. In other words $\pi' = Ranking(v \to p | S \setminus \{v\} \to \pi)$ for some $v \in S$ with $\pi(v) \in \{1, k\}$ and $p \in \{1/2, 5/2, k-3/2, k+1/2\}$. Observe that this is equivalent to ordinary fragility for $k \leq 3$.

The extension to weak fragility requires replacing the two Lemmas that use fragility, namely Lemmas 11 and 18. The new versions are identical except for constants. Other constants in the algorithm and other parts of the proof need to be adjusted accordingly.

Lemma 30 (Weak fragile version of Lemma 11)**.** *Let c be weakly fragile and $k \geq 3$. Let σ be an ordering of V, $v \in V$ be a vertex and $p, p' \in \mathbb{R}$. Let B be the set of vertices (excluding v) between p and p' in σ. Then $b(\sigma, v, p) + b(\sigma, v, p') \geq \frac{|B|}{(k-1)4^{k-2}}\binom{n-2}{k-2}$.*

Proof. By definition

$$b(\sigma, v, p) + b(\sigma, v, p') = \sum_{Q:\cdots} \left[c(Q \mapsto \sigma | v \mapsto p) + c(Q \mapsto \sigma | v \mapsto p')\right] \tag{11}$$

where the sum is over sets $Q \subseteq U \setminus \{v\}$ of $k-1$ vertices. Observe by *weak* fragility that the quantity in brackets in (11) is at least 1 for every Q that either has all $k-1$ vertices between p and p' in σ^2 or has one vertex between them and the remaining $k-2$ either all before or all after.

We consider two cases. If $|B| \geq |V|/2$ then the number of such Q is at least $\binom{|B|}{k-1} \geq \frac{|B|}{k-1}\binom{|B|-1}{k-2} \geq \frac{|B|}{(k-1)2^{k-2}}\binom{n-2}{k-2}$. If $|B| \leq |V|/2$ then at least $|V|/4$ vertices are either before or after hence the number of such Q is at least $|B|\binom{|V|/4}{k-2} \geq \frac{|B|}{2 \cdot 4^{k-2}}\binom{n-2}{k-2} \geq \frac{|B|}{(k-1) \cdot 4^{k-2}}\binom{n-2}{k-2}$ for sufficiently large n and $k \geq 3$. $\square$

For weak fragile problems we need to change the constant in the definition of $\bar{w}$. The new definition is $\bar{w}^{\sigma}_{uv} = w^{\sigma}_{uv} - \min(\frac{1}{10 \cdot 4^{k-2}}\binom{n-2}{k-2}, w^{\sigma}_{uv}, w^{\sigma}_{vu})$.

Lemma 31 (Weak fragile version of Lemma 18)**.** $\frac{1}{4^{k-2}}(1 - 2/10)\binom{n-2}{k-2} \leq \bar{w}^{\sigma^2}_{uv} + \bar{w}^{\sigma^2}_{vu} \leq (2 - \frac{2}{10 \cdot 4^{k-2}})\binom{n-2}{k-2}$*, i.e. $\bar{w}^{\sigma^2}$ is a weighted FAST instance.*

Proof. We prove the more interesting lower-bound and leave the straightforward proof of the upper bound to the reader. Fix $u, v \in U$. We consider two cases.

If there are at least $|U|/2$ vertices between u and v in σ^2 then we note that by weak fragility every constraint $S \supseteq \{u, v\}$ with all vertices in S between u and v in σ^2 contributes at least 1 to $w_{uv} + w_{vu}$. Therefore $w_{uv} + w_{vu} \geq \binom{|U|/2}{k-2} \geq \frac{1}{2 \cdot 2^k}\binom{n-2}{k-2}$.

If there are at most $|U|/2$ vertices between u and v in σ^2 then consider constraints with all their vertices either all before or all after u and v. We note that by weak fragility each such constraint

$S \supseteq \{u, v\}$ contributes at least 1 to $w_{uv} + w_{vu}$. There are clearly at least $|V|/4$ vertices either before or after, hence at least $\binom{|V|/4}{k-2} \geq \frac{1}{4^{k-2}}\binom{n-2}{k-2}$ constraints.

We conclude that $w_{uv} + w_{vu} \geq \frac{1}{4^{k-2}}\binom{n-2}{k-2}$. The Lemma follows from the definition of $\bar{w}$. $\square$