# How to rank with few errors

Claire Kenyon-Mathieu Warren Schudy
Brown University Computer Science
{claire,ws}@cs.brown.edu

October 16, 2006

**Abstract**

Suppose you ran a chess tournament, everybody played everybody, and you wanted to use the results to rank everybody. Unless you were really lucky, the results would not be acyclic, so you could not just sort the players by who beat whom. A natural objective is to find a ranking that minimizes the number of upsets, where an upset is a pair of players where the player ranked lower on the ranking beats the player ranked higher. This problem is the NP-hard minimum feedback arc set problem on tournaments. We give a polynomial time approximation scheme (PTAS) for this problem. A simple weighted generalization gives a PTAS for Kemeny-Young rank aggregation.

## 1 Introduction

Suppose you ran a chess tournament, everybody played everybody, and you wanted to use the results to rank everybody. Unless you were really lucky, the results would not be acyclic, so you could not just sort the players by who beat whom. A natural objective is to find a ranking that minimizes the number of upsets, where an upset is a pair of players where the player ranked lower on the ranking beats the player ranked higher. This problem is the minimum feedback arc set (FAS) problem on tournaments.

For a general directed graph, the minimum feedback arc set (FAS) problem consists of removing or reversing the fewest number of edges to make it acyclic. One of the original NP-complete problemd by reduction from vertex cover [39], it is NP-hard to approximate better than 1.36... [20]. A special case is to restrict the input to be a tournament graph, i.e. a directed graph where every pair of points is connected by exactly one of the two possible directed edges (undirected version is the complete graph).

**Problem 1** (FAS Tournament)**.** [49, 57, 51] Input: A tournament with $n$ vertices given by $V$ and edge relation $E(x, y)$. Output: A total order $R(x, y)$ minimizing the number of disagreements $\sum_{\{x,y\}\in V} \mathbb{1}\left(R(x, y) \neq E(x, y)\right)$.

Unfortunately, even with this restriction, it was recently shown by Alon, building on work by Ailon, Charikar and Newman, that FAS is still NP-hard [1, 2] (see also [14]).

The following generalization of the minimum FAS problem is useful for many applications.

**Problem 2** (Weighted FAS Tournament)**.** Input: A positive integer $n$, set $V$ and weights $w_{ij} \geq 0$ satisfying $0 < b \leq w_{ij} + w_{ji} \leq 1$ for some constant $b$. Output: A total order $R(x, y)$ minimizing $C_R \equiv \sum_{\{x,y\} \in V} w_{xy} R(x, y) + w_{yx} R(y, x)$.

The FAS tournament problem is the special case of the weighted FAS tournament problem in which we define $w_{ij} = \mathbb{1}(\neg E(i, j))$.[1] In [1], the variant of weighted FAS tournament such that $b = 1$ is called *weighted FAS tournament with probability constraints*.

The FAS problems on tournaments has a long and rich history, starting in the early 1960s in combinatorics [49, 57] and statistics [51]. Much early work consisted in searching for the worst-case tournament, which led to studying random tournaments. This direction led to a series of papers, starting with Erdös and Moon and culminating with work by Fernandez de la Vega [23, 46, 47, 34, 52, 53, 26] (see also [3]).

The operations research community has also been interested in the problem. The integer programming formulation of the minimum feedback arc set problem is due to Tucker [56]. Given a directed graph, linear relaxations of the linear ordering polytope provide useful lower bounds to the value of the minimum feedback arc set. This has been the focus of interest in mathematical programming [29, 30, 35, 42] and such lower bounds were essential in the approximation algorithms designed in [1].

Applications include scheduling [27] and graph layout [54, 18] (see also [44, 5, 35]).

A related problem is rank aggregation. Frequently, one has access to several rankings of objects of some sort, such as search engine outputs, and desires to aggregate the input rankings into a single output ranking that is similar to all of the input rankings. This problem has an even longer history, since it was already studied in the context of voting by Borda [10] and Condorcet [13]. Rank aggregation is usually done by finding a ranking with the minimum *average distance* from the input rankings, for some notion of distance. There are three classic objective functions for quantifying the distance between two rankings. Each objective function provides the maximum likelihood estimator for a naïve Bayesian assumption about how the input rankings are generated from an assumed real ranking. (See [15] for a discussion).

The first distance function, Spearman's foot rule, is the sum over candidates of the absolute value of the difference between the rank of the candidate in the two rankings. Spearman's foot rule assumes exponentially distributed noise of the ranks. With this measure, rank aggregation reduces to minimum cost perfect matching [21]. If each candidate has a unique median rank in the input rankings, then the foot rule optimal ranking can be obtained by sorting by the median rank [25]. The second distance function, Spearman's rank correlation, replaces absolute value by squaring. It assumes normally (Gaussian) distributed noise of the ranks. Rank aggregation can then be solved by Borda count [10] (breaking ties arbitrarily) [38], which is equivalent to sorting the vertices by weighted in degree.

The third distance function, the Kendall tau distance, which is the one studied here, is the number of pairs of candidates that are in a different order in the two rankings. It assumes that

[1] The lower bound of $b$ on the sum of weights is required to make this problem a generalization of FAS on tournaments but not so general as to be equivalent to FAS on general graphs. This lower bound appears in our analysis in the proof of Claim 5. There may be other assumptions that would also support our analysis.

there is a constant probability that any pair of candidates will be reversed. Rank aggregation with a Kendall tau metric is called Kemeny-Young rank aggregation [40, 41]. It is a special case of weighted FAS-tournament where the weight on an edge is the fraction of the input rankings that prefer that order for its endpoints [1], and is also NP hard [21]. The Kemeny Young rank aggregation can be approximated by using the other objective functions. Spearman's rank correlation gives a 5-approximation [16], and the foot rule gives a 2-approximation [1].

More recently, the machine learning community has been interested in rank aggregation and minimum feedback arc sets. Cohen, Shapire and Singer [12] model the aggregation of several ranking experts such as search engines as weighted Kemeny Young rank aggregation. Kamishima and Akaho [37] and Brinker [11], given training data which consists of (attribute, ranking) pairs, want to learn a function from attributes to rankings. For example, attributes might be the age, race, and gender of a movie renter, with rankings indicating what new releases a person likes best; they learn a function for each pair of movies from attributes to a value in [0,1] that indicates the probability that each movie should be ranked first; this is a variant of rank aggregation. Hüllermeier and Fürnkranz [33] study a related problem, where the training data gives pairwise information that a particular attributes has one label in front of another. They also learn a function giving the probability that one label is in front of the other.

Dwork, Kumar, Naor, and Sivakumar [21, 22] describes a connection between the Kemeny Young objective function and the Condorcet criterion beloved in voting theory. Their work implies that the orderings which we construct satisfy the generalized Condorcet criterion. However, using Kemeny Young rank aggregation for a voting system has serious weaknesses: see [6, 7, 43] for examples.

The best previously known approximation algorithms for weighted or unweighted feedback arc set on tournaments achieve constant factor approximations [1, 16, 58], with best factors at 2.5 in the randomized setting for both problems [1]. The best factors for a deterministic setting are 3 and 5 for the unweighted and weighted cases respectively by Ailon, Charikar and Newman [1] and van Zuylen[58]. Prior algorithms and heuristics can be found in [24, 27, 50, 31, 36, 55, 44, 5, 8, 9, 32, 45, 48] for example.The best-known factors for rank aggregation are 4/3 for randomized algorithms [1] and 2 for deterministic algorithms [1, 58]. On the other hand, Arora, Frieze and Kaplan [4] and Frieze and Kannan [28] give polynomial-time approximation schemes for the complementary maximization problem.

In this paper, we prove the following result.

**Theorem 1** (PTAS)**.** *There is a randomized Polynomial-Time Approximation Scheme for weighted Minimum Feedback Arc Set on tournaments with probability constraints and for Kemeny Young rank aggregation. Given $\epsilon, \delta > 0$, the algorithm outputs an ordering whose cost, with probability at least $1-\delta$, is less than $(1+O(\epsilon)/b)OPT$. The running time is $\tilde{O}\left(n^8/(b\epsilon) + n^6/(b\epsilon)2^{(1/\epsilon)^{O(1/\epsilon)}}\right)$. The algorithm can be derandomized at the cost of increasing the running time by a factor of $n^{\tilde{O}(1/\epsilon^2)}$.*

Indeed, the only source of randomness in our randomized algorithm comes from the algorithm in [4, 28]. The algorithm in [28] can be derandomized in polynomial time, so our algorithm can also be derandomized (at the cost of a $n^{\tilde{O}((1/\epsilon)^2)}$ factor increase in runtime) yielding the claimed deterministic running time.

Our analysis uses an interesting and, as far as the authors are aware, novel technique. Although the core of our algorithm is deterministic, its analysis uses a randomly generated tree that is analogous to a recursion tree.[2] We associate an upper bound with each tree and prove that the *expected* upper bound is less than $(1+\epsilon)OPT$. Since the actual cost of the algorithm is less than the minimum of the random upper bounds which we construct, it is also less than the expectation, hence the result.

We note that Amit Agarwal has informed us that he recently obtained similar results independently of this work.

## 2 Algorithm

One technique to construct a good ordering consists in ordering a small random sample of the vertices, and using that order as a hint to decide where to place all the other vertices. This approach was the one taken in [4, 28] and leads to a PTAS for the complementary maximization problem, i.e. to an algorithm with small additive error.

**Theorem 2** (Small additive error)**.** *[4, 28] Let $OPT$ denote the cost of an optimal ordering. For any fixed $\beta, \eta > 0$, there exists a randomized polynomial time algorithm AddApprox that outputs an ordering whose cost, with probability at least $1-\eta$, is less than $OPT + \beta n^2$. The best-known running time is $n^2 + 2^{O(1/\beta^2)} \log \eta$ in [28]. The algorithm in [28] can be derandomized by exhaustively considering all possibilities for the random choices, yielding a deterministic PTAS.*

Our algorithm implicitly uses that technique, since it iteratively improves its current ordering by applying the algorithm from [4, 28] to subsets of vertices which are consecutive in the current ordering.

Another natural and elementary technique to iteratively improve the current ordering consists of taking a vertex out of the ordering and moving it back at a different position. Any total order $R$ can equivalently be defined by an ordering $\sigma : V \to \{1, \dots, n\}$ such that $\sigma(x)$ is the position of vertex $x$: $\sigma(x) < \sigma(y)$ iff $R(x, y)$.[3]

**Definition 3.** A *single vertex move*, given a ordering $\pi$, a vertex $x$ and a position $i$, consists of taking $x$ out of $\pi$ and putting it back in in position $i$.

We are now ready to describe the PTAS. Assume that we are given an error parameter $\epsilon > 0$ and probability of failure $\delta > 0$. Recall from Problem 2 that $b$ is a lower bound on $w_{xy} + w_{yx}$ for every pair $\{x, y\}$.

**Algorithm 1. Input:** A weighted tournament with probability constraints.
**Preprocessing:** Round each weight to the nearest integer multiple of $\epsilon b/n^2$.
**Initialization:** Start from an arbitrary ordering $\pi$.
**While** there exists a cost-decreasing move, do that move. The two types of moves are:

---

[2]In early versions of our algorithm, it *was* a recursion tree.

[3]One could instead arbitrarily number the vertices and use permutations instead of orderings. We choose not to do this to make it clear that we do not use a total order on $V$.

1. **Single vertex moves.** Choose a vertex $x$ and a rank $j$, take $x$ out of the ordering $\pi$ and insert it back in so that its rank is $j$.

2. **Additive approximation.** Choose two integers $i < j$; let $U$ be the set of vertices whose current ranks are in $[i, j]$. Execute algorithm AddApprox on $U$, with $\beta = \alpha_{min}\epsilon^3$ and $\eta = \delta/n^4$, where $\alpha_{min} = 9^{-\log_{3/2}(1/\epsilon^2)/\epsilon} = \epsilon^{O(1/\epsilon)}$. Let $\pi'_U$ denote the result. Replace the restriction $\pi_U$ of $\pi$ to $U$ by $\pi'_U$.

**Output** $\pi$.

### Commentary

The doubly exponential behavior of the runtime of this algorithm with respect to $1/\epsilon$ is prohibitive in practice. The main algorithmic idea is to do a local search where one of the moves is running the be additive approximation algorithm on an interval of vertices in the current order. This idea may give reasonable heuristics in practice.

# 3 Analysis

## 3.1 Running time

The preprocessing step ensures that the edge weights are integer multiples of $b\epsilon/n^2$, so the cost decreases by at least $b\epsilon/n^2$ at each iteration. Since the cost is always between $n^2$ and zero, the total number of iterations is bounded by $n^4/(b\epsilon)$. Each iteration consists of $O(n^2)$ calls to the additive error and single vertex move algorithms. Thus the runtime is $O(n^6/(b\epsilon) f(n, (1/\epsilon)^{O(1/\epsilon)}) \log(n^4/\delta))$, where $f(n, \beta)$ is the time required for the additive approximation algorithm to run on the problem of size $n$ with error parameter $\beta$. The best-known additive approximation run time $f(n, \beta)$ is $n^2 + 2^{O(1/\beta^2)}$ in [28].

## 3.2 Quality of approximation

We first prove a reduction to the rounded problem: the rounding is a minor issue.

**Lemma 4** (Rounding)**.** *Let $C$ denote the cost function with the original weights and $\tilde{C}$ denote the cost function with the rounded weights, and $\tilde{OPT} = \min_\rho \tilde{C}(\rho)$ denote the value of the optimal solution to the problem with rounded weights. Assume that $\tilde{C}(\pi) \leq (1 + \epsilon)\tilde{OPT}$. Then $C(\pi) \leq (1 + 3\epsilon)OPT$.*

*Proof.* Appendix. □

Thanks to Lemma 4, we can analyze the rounded weights as if they were the original weights. For the remainder of the paper we analyze the operation of the algorithm on the rounded graph as if it were the original graph.[4]

[4] Put another way, without loss of generality we can assume that the input weights are integer multiples of $\epsilon/n^2$.

### 3.2.1 Proof of Theorem 1: outline

**Intuition.** Here is the intuitive story behind that algorithm. It is at least as good as a divide-and-conquer algorithm that starts with an ordering which is a local optimum with respect to moving single vertices, and repeatedly cuts it up using divide-and-conquer, until the intervals are small enough that the algorithm with small additive error is helpful.

The interesting case is when the cost of the optimal ordering $\sigma$ is quite low. Assume that you start with an ordering $\pi$ whose cost is within a constant factor of optimal. Then $\sigma$ and $\pi$ cannot be very different: Typical possibilities would be that they differ perhaps by a few long displacements, or perhaps by many short displacements. Moreover, by local optimality, $\pi$ cannot be improved by moving a vertex $x$ from position $\pi(x)$ to position $\sigma(x)$, and $\sigma$ cannot be improved by moving $x$ from position $\sigma(x)$ to position $\pi(x)$: this seems to almost require $\sigma(x)$ and $\pi(x)$ to be close. So, assume that all displacements are short. Then divide-and-conquer makes sense, since intuitively, getting a few vertices on the wrong side will not hurt much. The base case of this divide-and-conquer approach would be when the interval size is not much more than the average vertex displacement, and then one can just apply the maximization PTAS.

One flaw of the intuitive reasoning arguing that $x$ should have similar positions in $\rho$ and in $\sigma$ is that complications occur when the vertices positioned between $\pi(x)$ and $\sigma(x)$ are not the same in the $\pi$ ordering as in the $\sigma$ ordering. Such complications occur when some vertex $y$ has $\sigma(y)$, but not $\pi(y)$, between $\sigma(x)$ and $\pi(x)$. In such situations we say that $x$ and $y$ "cross", and analyzing such crossings is a big part of our work.

Let $\sigma$ be an optimal ordering of the vertices. (Notice that, once restricted to a proper subset of the vertices, $\sigma$ need not be the optimal ordering for that subset). Let $\pi$ denote the output ordering. We will construct a certain random hierarchical decomposition of $V$ into subsets. Let $(\alpha, B = (\zeta_u))$ be the set of random numbers used in the construction. Each subset in the decomposition is of the form $V_{k,\ell} = \{x : k \leq \pi(x) \leq \ell\}$. The hierarchical decomposition yields a certain upper bound on the cost of $\pi$. We will prove that in expectation (where the expectation is over $B$ and $\alpha$), this upper bound is less than $C(\sigma)(1 + O(\epsilon))$. This implies that $C(\pi) \leq C(\sigma)(1 + O(\epsilon))$, hence the Theorem.

Here is the construction of the decomposition. We use the following quantity, based on the foot rule distance between two orderings : for any set of vertices $u$ let $F_u = \sum_{x \in u} |\pi(x) - \sigma(x)|$. The following claim allow us to bound the extra cost $C_V(\pi) - C_V(\sigma)$ by $\epsilon F_V$ instead of $\epsilon C_V(\sigma)$.

**Claim 5** (Displacements)**.** *If $\sigma$ and $\pi$ are two orderings, then $F_V = \sum_x |\pi(x) - \sigma(x)|$ is less than $(2/b)(C_V(\sigma) + C_V(\pi))$.*

*Proof.* Appendix ☐

**Algorithm 2.** Input: orderings $\pi, \sigma$
Output: a hierarchical decomposition of $V$
Initialization:

1. Let $\alpha = 9^{-r}$, where $r$ is an integer uniformly distributed in $\{0, 1, \ldots, \log_{3/2}(1/\epsilon^2)/\epsilon\}$.

2. Return the tree built by Process$(1, n)$

Let Process$(i, j)$ = begin
  Let $V_{i,j} = \{x : i \le \pi(x) \le j\}$
  if $F_{V_{i,j}} < \alpha\epsilon^2 |V_{i,j}|^2$ and $i < j$,
    Let $\zeta$ be a random real number from $[1/3, 2/3]$.
    Let $k$ be such that $|V_{i,k}| = \lfloor \zeta |V_{i,j}| \rfloor$
    Put the subsets $V_{i,k}$ and $V_{k+1,j}$ in the decomposition, and define them as the left child and right child of the node labelled $V_{i,j}$.
    left child $\leftarrow$ Process$(i, k)$ and right child $\leftarrow$ Process$(k + 1, j)$
end

Note that $\alpha \in [\alpha_{min}, 1]$ with $\alpha_{min} = 9^{-\log_{3/2}(1/\epsilon^2)/\epsilon}$. The choice of $\alpha$ and of the random $\zeta$'s determines the decomposition. Let $B = (\zeta_u)$ be a family of random variables, indexed by nodes of an infinite binary tree: here $u$ is a word in the language $\{\text{left child}, \text{right child}\}^*$. The $\zeta_u$ are i.i.d.r.v's drawn from the uniform distribution over $[1/3, 2/3]$. Thus, for a given $\pi$, the tree decomposition is determined by $(B, \alpha)$. Given any ordering $\pi$, let $w_e(\pi)$ denote the cost of edge $e$ under ordering $\pi$: for an edge $e = \{x, y\}$, either $w_{xy}$ if $\pi(x) < \pi(y)$ or $w_{yx}$ otherwise. The optimal ordering minimizes $C_V(\pi)$, where for a subset of vertices $u$, we write $C_u(\pi) = \sum_{\{x,y\} \subseteq u} w_{xy}(\pi)$.

Consider the event $E_0$ stating that "During execution of the algorithm, every call to algorithm AddApprox yields a result within the stated bounds." Each call fails with probability at most $\eta = \delta/n^4$; there are at most $n^4$ such calls, so event $E_0$ has probability at least $1 - \delta$ all succeed. We hereafter assume that event $E_0$ holds and do our analysis conditionial on $E_0$. The following claim motivates the stopping condition of algorithm 2.

**Claim 6** (Leaf nodes)**.** *Let $\sigma$ be an optimal ordering. Let $v$ be a leaf of the recursion tree. If event $E_0$ holds, then $C_v(\pi) - C_v(\sigma) \le \epsilon F_v$.*

*Proof.* If $|v| = 1$ then the claim is trivial. Otherwise, using Theorem 2 and the definition of tree leaves:

$$C_v(\pi) - C_v(\sigma) \le C_v(\pi) - \mathrm{OPT}_v \le \alpha_{min}\epsilon^3 |v|^2 \le (\alpha_{min}\epsilon/\alpha) F_v \le \epsilon F_v,$$

where the last inequality follows from $\alpha \ge \alpha_{min}$. □

We will bound the mistakes made between the leaves and the root by splitting the mistakes into two parts and bounding each separately. Let $f_\pi(e) = w_e(\pi^R) - w_e(\pi)$. Here $\pi^R$ denote the reverse ordering of $\pi$, such that $\pi^R(x) = n - \pi(x)$. Thus $f_\pi(e)$ denotes the cost, for edge $e = \{x, y\}$ of switching the relative order of $x$ and $y$. For any subset $u$ of the vertices, let $g_u^\pi(x, i)$ denote the cost due to edges between $x$ and vertices in $u$ and incurred by removing $x$ from the ordering, and putting it back in so that its rank is $i$:

$$g_u^\pi(x, i) = \begin{cases} \sum_{z \in u : \pi(x) < \pi(z) \le i} f_\pi(e) & \text{if } \pi(x) < i \\ \sum_{z \in u : i \le \pi(z) < \pi(x)} f_\pi(e) & \text{if } \pi(x) \ge i, \end{cases}$$

Note that an ordering $\pi$ is locally minimal with respect to single vertex moves iff for every vertex $x$ and position $i$, we have $g_V^\pi(x, i) \ge 0$. Given a subset $u$ of the vertices, let $T_u = \sum_{x \in u} -g_u^\pi(x, \sigma(x))$.

To each vertex $x$, associate the interval $I_x = [\min(\pi(x), \sigma(x)), \max(\pi(x), \sigma(x))]$. Say that a pair of vertices $x$ and $y$ are *crossing* if the two intervals $I_x$ and $I_y$ intersect but neither is contained in the other interval.

**Lemma 7** (Local). *We have: $C_V(\pi) - C_V(\sigma) = T_V + \Phi_V$, where $\Phi_V = \sum_{x,y \text{ crossing pair}} \delta_{xy} f_\pi(x,y)$, and*

$$\delta_{xy} = \begin{cases} -1 & \text{if } \min(\pi(x), \pi(y)) < \sigma(x), \sigma(y) < \max(\pi(x), \pi(y)) \\ +1 & \text{otherwise.} \end{cases}$$

*Proof.* First note that $C(\pi) - C(\sigma) = \sum_{x,y:\pi(x)<\pi(y) \text{ and } \sigma(x)>\sigma(y)} f_\pi(x,y)$. By definition, we have $T_V = \sum_x \sum_{y:\pi(y) \text{ between } \pi(x) \text{ and } \sigma(x)} f_\pi(x,y)$. Note that if $I_x$ and $I_y$ are disjoint then the pair $\{x,y\}$ does not appear in either sum. If one of $I_x, I_y$ is contained in the other interval, then the pair $\{x,y\}$ appears exactly once in both sums. Thus

$$C_V(\pi) - C_V(\sigma) - T_V = \sum_{x,y \text{ crossing pair}} \delta_{xy} f_\pi(x,y),$$

where $\delta_{xy}$ depends on the relative order of $\pi(x), \pi(y), \sigma(x)$ and $\sigma(y)$. A short case-by-case analysis concludes the proof.

□

Lemma 7 allows us to analyze the increase from the leaves to the root of $T$ and $\Phi$ separately. The following two lemmas are proved later.

**Lemma 8.** *For any $\sigma$ and $\pi$ locally optimal with respect to single vertex moves, for a random tree construction defined by a random choice of $(B, \alpha)$, we have*

$$T_V \leq 11\epsilon F_V + E_{B,\alpha}\left[\sum_{v \text{ leaf}} T_v\right].$$

**Lemma 9.** *Let $\gamma = \sqrt{5}/3$. For any $\pi, \sigma$ and for any $\alpha$, for a random tree construction defined by a random choice of $(B)$, we have*

$$\Phi_V \leq 72\frac{\gamma}{1-\gamma}\epsilon F_V + E_B\left[\sum_{v \text{ leaf}} \Phi_v\right].$$

Now we can prove Theorem 1: Recall that $\sigma$ is an optimal ordering of the vertices and $\pi$ is the ordering output by Algorithm 1. Since Lemma 9 is true for every $\alpha$, it is also true in expectation over $\alpha$. Adding to Lemma 8 yields:

$$T_V + \Phi_V \leq (11 + 72\frac{\gamma}{1-\gamma})\epsilon F_V + E_{B,\alpha}\left[\sum_{v \text{ leaf}} (T_v + \Phi_v)\right].$$

Apply Lemma 7 to both sides, yielding:

$$C_V(\pi) - C_V(\sigma) \leq (11 + 72\frac{\gamma}{1-\gamma})\epsilon F_V + E_{B,\alpha}\left[\sum_{v \text{ leaf}} (C_v(\pi) - C_v(\sigma))\right]$$

By Claim 6, whenever event $E_0$ holds, the expectation on the right-hand side is bounded above by $\sum_{v \text{ leaf}} \epsilon F_v = \epsilon F_V$, which is independant of $B, \alpha$ so the expectation can be dropped. Therefore (conditioned on event $E_0$ being true):

$$C_V(\pi) - C_V(\sigma) \leq (11 + 72\frac{\gamma}{1-\gamma} + 1)\epsilon F_V$$

Using Claim 5:

$$C_V(\pi) - C_V(\sigma) \leq (72\frac{\gamma}{1-\gamma} + 12)\epsilon\frac{2}{b}(C_V(\pi) + C_V(\sigma)).$$

Solving for $C_V(\pi)$ and remembering that $\gamma = \sqrt{5}/3 = 0.745...$:

$$C_V(\pi) \leq \frac{1 + (72\gamma/(1-\gamma) + 12)\epsilon}{1 - (72\gamma/(1-\gamma) + 12)\epsilon} C_V(\sigma) = (1 + O(\epsilon))C_V(\sigma),$$

which concludes the proof.

It only remains to prove Lemmas 9 and 8.

### 3.2.2 Proof of Lemma 9

**Notation 10.** $\psi_u(k)$ (resp. $\psi_u^R(k)$)denote the number of vertices in $u$ that are strictly after (resp. before) position $k$ in the $\pi$ ordering but before (resp. strictly after) position $k$ in the $\sigma$ ordering: $\psi_u = |\{\, x \in u |\ (\sigma_(x) \leq k < \pi(x)\,\}|$.

**Lemma 11** (Core)**.** *Given a set of vertices $S = \{x, \pi(x) \in [\ell, r]\}$, let $\zeta$ be a random number from $[1/3, 2/3]$ and $k = \lfloor \ell + \zeta(1/3)(r - \ell)\rfloor$. Let $L = \{x \in S, \pi(x) \leq k\}$ and $R = S \setminus L$. Then:*

$$\Phi_S \leq E_\zeta[\Phi_L + \Phi_R] + \frac{24}{|S|}\sum_{x \in S} |\sigma(x) - \pi(x)|\psi_S^*.$$

*where* $\psi_S^* = \max_{k \in S} \max(\psi_S(k), \psi_S^R(k))$

*Proof.*

$$\begin{aligned}
\Phi_S &= \sum_{x,y \in S, x,y \text{ crossing pair}} \delta_{xy} f_\pi(x,y). \\
\Phi_L &= \sum_{x,y \in L, x,y \text{ crossing pair}} \delta_{xy} f_\pi(x,y). \\
\Phi_R &= \sum_{x,y \in R, x,y \text{ crossing pair}} \delta_{xy} f_\pi(x,y).
\end{aligned}$$

Since $|\delta_{xy}| \leq 1$ and $|f_\pi(x,y)| \leq 1$, we can write

$$\Phi_S - \Phi_L - \Phi_R \leq |\{(x,y) \in L_k \times R_k \cup R_k \times L_k, \text{ crossing pair}\}|,$$

where we write $L = L_k$ and $R = R_k$ to make the dependency in $k$ explicit.

Since $k$ is distributed uniformly at random in the middle third of interval $[\ell, r]$, we have:

$$E_\zeta[\Phi_S - \Phi_L - \Phi_R] \leq \frac{3}{|S|}|\{(x,y,k) : (x,y) \in L_k \times R_k \cup R_k \times L_k, \text{ crossing pair}\}|.$$

In such a triple $(x,y,k)$, we must have that $k$ either belongs to $[\min(\pi(x),\sigma(x)),\max(\pi(x),\sigma(x)]$ or to $[\min(\pi(y),\sigma(y)),\max(\pi(y),\sigma(y)]$. Moreover, in order for $\{x,y\}$ to be a crossing pair, it must be that $y$ cross over either $\pi(x)$ or over $\sigma(x)$, so $y$ is counted in $\psi_S(\pi(x))$ or in $\psi_S^R(\pi(x))$ or in $\psi_S(\sigma(x))$ or in $\psi_S^R(\sigma(x))$. Taking the union of these cases yields the lemma.

□

**Lemma 12.** *For any $S$ and $k$, we have: $\psi_S(k) \leq F_S^{1/2}$ and $\psi_S^R(k) \leq F_S^{1/2}$.*

*Proof Sketch.* $i \mapsto \psi_S(i)$ is a a function which equals 0 at both endpoints of the interval, which is 1-Lipschitz (it changes by at most 1 when going from $i$ to $i+1$), and whose integral is bounded by $F_S$. Together, it is easy to see that these observations imply that the maximum value of this function cannot be more than $\sqrt{F_S}$. See Appendix for a full proof. □

**Claim 13** (Algebra)**.** *Let $\gamma = \sqrt{5}/3$. If $0 \leq F_1 \leq F$, $s > 0$, and $\zeta \in [1/3, 2/3]$, then:*

$$\frac{F^{3/2}}{s} < \gamma\left(\frac{F_1^{3/2}}{\zeta s} + \frac{(F-F_1)^{3/2}}{(1-\zeta)s}\right).$$

*Proof.* Appendix □

Combining the above lemmas enables us to express $\Phi_V$ solely as a function of the leaves of the tree.

**Lemma 14** (Inductive)**.** *For every $\alpha \in [\alpha_{min}, 1]$, $\Phi_V \leq E_B\left[\sum_{v \text{ leaf}}[72\frac{\gamma}{1-\gamma}\epsilon F_v + \Phi_v] \text{ given } \alpha\right]$.*

*Proof.* Let $\alpha \in [\alpha_{min}, 1]$ be fixed. Given $\alpha$, we first prove by induction that

$$\Phi_V \leq E_B\left[\sum_{v \text{ internal}} 24\frac{F_v^{3/2}}{|v|} + \sum_{v \text{ leaf}} \Phi_v\right]. \tag{1}$$

If $V$ is a leaf, then the statement is true. Else, let $\zeta$ be the random variable used at the root to decompose $V$ into $V_1$ and $V_2$. Let $X_\zeta = \Phi_V - \Phi_{V_1} - \Phi_{V_2}$. Let $B_1$ be the random sequence used to construct the decomposition of the left child, and $B_2$ be the random sequence used to construct the decomposition of the right child, so that $B = (\zeta, B_1, B_2)$. Since $X_\zeta$ is independent of $B_1, B_2$, the decomposition of $V_1$ is independent of $B_2$, and the decomposition of $V_2$ is independent of $B_1$, we have:

$$\Phi_V = E_B(X_\zeta + \Phi_{V_1} + \Phi_{V_2}) = E_\zeta(X_\zeta) + E_\zeta(E_{B_1}(\Phi_{V_1})) + E_\zeta(E_{B_2}(\Phi_{V_2})).$$

By Lemma 11 and Lemma 12 we have $E_\zeta(X_\zeta) \leq 24F_V^{3/2}/|V|$. For each value of $\zeta$, we can apply induction to $V_1$ and $V_2$. Summing proves Equation 1.

The rest of the proof is deterministic. Fix a tree decomposition and analyze the right hand side of Equation 1. For every internal node $v$, by Claim 13 we have: $F_v^{3/2}/|v| \leq \gamma(F_{v_1}^{3/2}/|v_1| + F_{v_2}^{3/2}/|v_2|)$. Summing over the recursion tree, we obtain:

$$\sum_{u \text{ internal}} \frac{F_u^{3/2}}{|u|} \leq \sum_{u \text{ leaf}} \frac{F_u^{3/2}}{|u|}(\gamma + \gamma^2 + \cdots) \leq \frac{\gamma}{1-\gamma} \sum_{u \text{ leaf}} \frac{F_u^{3/2}}{|u|}.$$

If $u$ is a leaf, then let $v$ be the parent of $u$.[5] The parent is not a leaf, so by the stopping condition $\sqrt{F_v}/|v| \leq \alpha\epsilon \leq \epsilon$. Thus:

$$\frac{F_u^{3/2}}{|u|} = F_u \frac{\sqrt{F_u}}{|u|} \leq F_u \frac{\sqrt{F_v}}{|v|}\frac{|v|}{|u|} \leq 3F_u \frac{\sqrt{F_v}}{|v|} \leq 3\epsilon F_u.$$

Summing yields the lemma. □

Combining lemma 14 with the simple fact that $\sum_{v \text{ leaf}} F_v = F_V$ proves lemma 9.

### 3.2.3 Proof of Lemma 8

In this part, we will prove that $T_V - E_{B,\alpha}(\sum_{v \text{ leaf}} T_v) \leq 11\epsilon F_V$, which implies Lemma 8. First we see that by definition of $T_V$ and $\{T_v\}$,

$$T_V - E_{B,\alpha}(\sum_{v \text{ leaf}} T_v) = E_{B,\alpha}\left[\sum_y \sum_{\substack{x\,:\,\pi(x) \text{ between } \pi(y) \text{ and } \sigma(y) \\ x \notin \text{ leaf containing } y}} f_\pi(x,y)\right].$$

Let $s(y)$ be the size of the leaf containing $y$. We can split the sum into several parts.

**Lemma 15** (Small Leaves)**.** *For any tree (determined by $B$ and $\alpha$), we have*

$$\sum_{y:|\sigma(y)-\pi(y)|>s(y)/\epsilon} \sum_{\substack{x\,:\,\pi(x) \textit{ between } \pi(y) \textit{ and } \sigma(y) \\ x \notin \textit{ leaf containing } y}} -f_\pi(x,y) \leq \epsilon F_V. \tag{2}$$

[5] If $u$ is the root, no parent is available but in that case the lemma is trivially true anyway.

*Proof.* No single vertex move can improve $\pi$, so $\sum_{x:\pi(x) \text{ between } \pi(y) \text{ and } \sigma(y)} -f_\pi(x,y) \leq 0$. Thus the left hand side of Equation 2 can be bounded by

$$\sum_{y:|\sigma(y)-\rho(y)|>s(y)/\epsilon} \sum_{\substack{x\,:\,\pi(x) \text{ between } \pi(y) \text{ and } \sigma(y) \\ x\in \text{ leaf containing } y}} f_\pi(x,y) \leq \sum_{y:|\sigma(y)-\rho(y)|>s(y)/\epsilon} s(y).$$

This is at most $\sum_y \epsilon|\sigma(y)-\pi(y)| = \epsilon F_V$. $\square$

**Lemma 16** (Big Leaves)**.** *For any $\alpha$ and for a random $B$, we have:*

$$E_B\left[\sum_{y:|\sigma(y)-\pi(y)|<\epsilon s(y)} \sum_{\substack{x\,:\,\pi(x) \textit{ between } \pi(y) \textit{ and } \sigma(y) \\ x\notin \textit{ leaf}(y)}} -f_\pi(x,y) \quad \textit{given} \quad \alpha\right] \leq 9\epsilon F_V.$$

*Proof.* Let $\alpha$ be fixed. If $\pi(x)$ is between $\pi(y)$ and $\sigma(y)$, but $x$ is not in leaf$(y)$, then $\pi^{-1}(\sigma(y))$ is also not in leaf$(y)$. Thus we can bound the expression on the left hand side by the expectation (over the random tree construction) of

$$\sum_y |\sigma(y)-\pi(y)| \cdot \mathbb{1}\left(|\sigma(y)-\pi(y)| < \epsilon\cdot s(y) \text{ and } \pi^{-1}(\sigma(y)) \notin \text{ leaf}(y)\right).$$

We will now argue that for any vertex $y$, the event $E_1$ that "$|\sigma(y)-\pi(y)| < s(y)\epsilon$ and $\pi^{-1}(\sigma(y)) \notin$ leaf$(y)$" has probability at most $9\epsilon$ over the sequence $B$ of random choices defining the decomposition. Indeed, fix a vertex $y$ and let $\ell = |\sigma(y)-\pi(y)|$. Down the branch leading to the leaf of $y$, the random tree construction uses a certain sequence of random variables $\zeta_0, \zeta_1, \ldots$. Instead of stopping when the decomposition stopping condition is reached[6], we conceptually extend the construction until the first time $Z$ that the associated interval has size less than $\ell/\epsilon$. This defines a sequence of nested intervals $I_0 = [1,n]$, $I_1$, …, $I_Z$.

The probability of event $E_1$ is bounded by the probability that the last interval does not contain $\sigma(y)$, hence:

$$\Pr(E_1) \leq \sum_{i<Z} \Pr(\sigma(y) \notin I_{i+1} | \sigma(y) \in I_i).$$

The probability in the right hand side is that the random cutting point used by the construction falls between $\pi(y)$ and $\sigma(y)$. There are only $\ell$ possibilities, so the probability is at most $3\ell/|I_i|$. For $i = Z-1$, by definition of $Z$ we have $|I_i| > \ell/\epsilon$. For $i \leq Z-1$, we can write

$$|I_i| = \frac{|I_i|}{|I_{i+1}|}\frac{|I_{i+1}}{|I_{i+2}|}\cdots\frac{|I_{Z-2}|}{|I_{Z-1}|}|I_{Z-1}| \geq (3/2)^{Z-i-1}(\ell/\epsilon).$$

[6]That would introduce some conditioning.

So

$$\Pr(E_1) \le \sum_{i=0}^{Z-1} \frac{3\ell}{(3/2)^{Z-i-1}(\ell/\epsilon)} \le 9\epsilon.$$

Summing yields the lemma.

□

**Lemma 17** (Intermediate Leaves). *For any set $B$ and for a random $\alpha$, we have*

$$E_\alpha \left[ \sum_{y:\epsilon s(y) \le |\sigma(y)-\pi(y)| \le s(y)/\epsilon} \quad \sum_{\substack{x\,:\,\pi(x) \text{ between } \pi(y) \text{ and } \sigma(y) \\ x \notin \text{ leaf}(y)}} -f_\pi(x,y)) \right] \le \epsilon F_V.$$

*Proof.* Let $B = (\zeta_u)$ be fixed.

As in the beginning of the proof of Lemma 16, we can bound the expression on the left hand side by the expectation (over the random choice of $\alpha$) of

$$\sum_y \begin{cases} 0 & \text{if } s(y) = n \\ \sum_{x:\pi(x) \text{ between } \pi(y) \text{ and } \sigma(y)} -f_\pi(x,y) & \text{if } s(y) = 1 \\ |\sigma(y)-\pi(y)| \cdot \mathbb{1}\left(\epsilon \cdot s(y) < |\sigma(y)-\pi(y)| < (1/\epsilon)\cdot s(y) \text{ and } \pi^{-1}(\sigma(y)) \notin \text{ leaf}(y)\right) & \text{otherwise.} \end{cases}$$

We will again argue that for any $y$, the event $E_2$ that "$\epsilon \cdot s(y) < |\sigma(y)-\pi(y)| < (1/\epsilon)\cdot s(y)$ and $\pi^{-1}(\sigma(y)) \notin$ leaf$(y)$ and $1 < s(y) < n$" has low probability; but here for the first time, the probabilistic space is over the random definition of $\alpha$.

Recall that $\alpha = 9^{-r}$, with $r$ chosen uniformly at random in $\{0, 1, \ldots, \log_{3/2}(1/\epsilon^2)/\epsilon\}$.

Fix a vertex $y$ and let $\ell = |\sigma(y)-\pi(y)|$. Go down the branch leading to $y$. Let $v_0, v_1, v_2, \ldots$ be the sequence of nodes on that branch, and let $a_i = F_{v_i}/|v_i|^2$. Every time you go down from one node $v_i$ to a child $v_{i+1}$, since $F_{v_{i+1}} \le F_{v_i}$ and $|v_{i+1}| \ge (1/3)|v_i|$, you have $a_{i+1} \le 9a_i$. By definition of the recursion tree, the leaf of $y$ is the *first* node $v_t$ along this branch where $a_t \ge \alpha\epsilon^2$ or which has size 1. Thus $a_t \le 9a_{t-1} < 9\alpha\epsilon^2$, and so, for any $\alpha' \ge 9\alpha$, $t$ is an internal node (if it has size greater than 1). Therefore a node $t$ of size greater than 1 can be a leaf for only one value of $\alpha = 9^{-r}$.

Consider the node $u$ of size 1. Even if $\ell < 1/\epsilon$, then whenever $\alpha$ is such that $u$ is the leaf of $y$, by local optimality with respect to single vertex moves, $y$ contributes $\le 0$ to the sum.

Consider the nodes $u$ of size strictly between 1 and $n$, and such that $\epsilon\ell < |u| < \ell/\epsilon$: there are at most $\log_{3/2}(1/\epsilon^2)$ such nodes, hence event $E_2$ has probability at most $\log_{3/2}(1/\epsilon^2)/(\log_{3/2}(1/\epsilon^2)/\epsilon) = \epsilon$.

Summing concludes the proof.

□

# 4 Conclusions

We note that the actual definition of ordering $\pi$ as being the output of Algorithm 1 only came into play in two places: in the proof of Claim 6, where we used the fact that Algorithm AddApprox has small additive error, and in the proof of Lemma 15, where we used the fact that $\pi$ could not be improved by doing single vertex moves. The assumption of the graph is a tournament (actually weighted generalization thereof) occurs only in the proof of Claim 5.

### Acknowledgements

The authors wish to thank Eli Upfal for useful discussions.

# References

[1] Nir Ailon, Moses Charikar, Alantha Newman: Aggregating inconsistent information: ranking and clustering. ACM Symposium on Theory of Computing (STOC) 2005: 684-693.

[2] N. Alon, Ranking tournaments, SIAM J. Discrete Math. 20 (2006), 137-142.

[3] N. Alon, J. Spencer. The probabilistic method, second edition. Wiley-Interscience. 2000.

[4] Sanjeev Arora, Alan Frieze, and Haim Kaplan. A New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangement Problems. IEEE Symposium on Foundations of Computer Science (FOCS) 1996: 24-33.

[5] F.B.Baker and L.J.Hubert, Applications of combinatorial programming to data analysis: seriation using asymmetric proximity measures, British J. Math. Statis. Psych. 30 (1977) 154-164.

[6] John J. Bartholdi, J. A. Orlin. Single transferable vote resists strategic voting. Social Choice and Welfare 8:341-354 (1991).

[7] JJ Bartholdi, CA Tovey, MA Trick . The computational difficulty of manipulating an election. Social Choice and Welfare, 6, 3, 1989.

[8] J.C. Bermond, Ordres à distance minimum d'un tournoi et graphes partiels sans circuits maximaux, Math. Sci. Humaines 37 (1972), 5-25.

[9] J.C. Bermond and Yves Kodratoff, Une heuristique pour le calcul de l'indice de transitivité d'un tournoi, RAIRO Inform Theory 10(1976) 83-92.

[10] J. Borda. Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences, 1781.

[11] Brinker K. Active learning of label ranking functions. Proceedings of the twenty-first international conference on Machine learning (ICML). ACM, 2004.

[12] William W. Cohen, Robert E. Schapire and Yoram Singer. Learning to order things. Proceedings of the 1997 conference on Advances in neural information processing systems (NIPS): 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] Conitzer, V. Computing Slater Rankings Using Similarities Among Candidates. Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06), Boston, MA, USA, 613-619. 2006.

[15] Vincent Conitzer and Tuomas Sandholm. Common Voting Rules as Maximum Likelihood Estimators. In Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), pp. 145-152, Edinburgh, Scotland, UK, 2005.

[16] D. Coppersmith, Lisa Fleischer, Atri Rudra: Ordering by weighted number of wins gives a good ranking for weighted tournaments. ACM-SIAM Symposium on Discrete Algorithms (SODA) 2006: 776-782.

[17] Andrew J. Davenport, Jayant Kalagnanam: A Computational Study of the Kemeny Rule for Preference Aggregation. AAAI 2004: 697-702.

[18] C. Demetrescu and I. Finocchi. Break the "right" cycles and get the "best" drawing. In B.E. Moret and A.V. Goldberg, editors, Proceedings of the 2nd International Workshop on Algorithmic Engineering and Experiments (ALENEX), 171-182, 2000.

[19] P. Diaconis and R. Graham. Spearman's footrule as a measure of disarray. J. of the Royal Statistical Society, 39(2), pages 262–268, 1977.

[20] I. Dinur and S. Safra. The importance of being biased. In Proceedings on 34th Annual ACM Symposium on Theory of Computing, pages 33-42, 2002.

[21] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation revisited. Manuscript. 2001.

[22] Cynthia Dwork, Ravi Kumar, Moni Naor, D. Sivakumar. Rank Aggregation Methods for the Web. WWW10. 2001.

[23] P. Erdös and J.W.Moon, On sets on consistent arcs in tournaments, Canadian Mathematical Bulletin 8 (1965) 269-271.

[24] G. Even, J. Naor, S. Rao, B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 62-71, 1995.

[25] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. Proceedings of the 2003 SIGMOD.

[26] Fernandez de la Vega, W. On the maximal cardinality of a consistent set of arcs in a random tournament, J. Combinatorial Theory, Ser. B 35:328-332 (1983).

[27] M.M.Flood. Exact and Heuristic Algorithms for the Weighted Feedback Arc Set Problem: a Special Case of the Skew-Symmetric Quadratic Assignment Problem. Networks, 20:1-23, 1990.

[28] Alan M. Frieze, Ravi Kannan: Quick Approximation to Matrices and Applications. Combinatorica 19(2): 175-220 (1999).

[29] M. Grötschel, M. Jünger, G. Reinelt. Facets of the linear ordering polytope. Math. Programming 33, 1985, 43-60.

[30] M. Grötschel, M. Jünger, G. Reinelt. On the acyclic subgraph polytope. Math. Programming 33, 1985, 43-60.

[31] S.L. Hakimi, On the degrees of vertices of a directed graph, J. Franklin Inst. 279 (1965) 290-308.

[32] L. Hubert, Seriation using asymmetric proximity measures, British J. Math Statist. Psych. 29 (1976) 32-52.

[33] E. Hüllermeier, J. Fürnkranz. Ranking by pairwise comparison: a note on risk minimization. 2004 IEEE international conference on fuzzy systems 97- 102 vol.1.

[34] H.A. Jung, On subgraphs without cycles in tournaments, Combinatorial Theory and its Applications II, North Holland (1970) 675-677.

[35] M. Jünger, Polyhedral combinatorics and the Acyclic Subdigraph Problem (Heldermann Verlag, Berlin, 1985).

[36] T. Kamae, Notes on a minimum feedback arc set, IEEE Trans. on Circuit Theory, 14 (1967) 78-79.

[37] Toshihiro Kamishima, Shotaro Akaho. Learning from Order Examples Proceedings of The IEEE International Conference on Data Mining. (2002)

[38] Toshihiro Kamishima, Jun Fujiki. Clustering Orders. Proc. of The 6th Int'l Conf. on Discovery Science. pp 194-207. 2003.

[39] Karp, R., 1972. Reducibility among combinatorial problems. In: Miller, R. E., Thatcher, J. W. (Eds.), Complexity of Computer Computations. Plenum Press, NY, pp. 85–103.

[40] J. Kemeny. Mathematics without numbers. Daedalus, 88:571-591, 1959.

[41] J. Kemeny and J. Snell. Mathematical models in the social sciences. Blaisdell, New York, 1962. Reprinted by MIT press, Cambridge, 1972.

[42] B. Korte, Approximative algorithms for discrete optimization problems, Ann. Discrete Math 4 (1979), 85-120.

[43] Lars-Gunnar Svensson. The Proof of the Gibbard-Satterthwaite Theorem Revisited. Working Paper No 1999:1, Department of Economics, Lund University. 1999. http://swopec.hhs.se/lunewp/papers/lunewp1999_001.pdf

[44] A. Lempel and I. Cederbaum. Minimum feedback arc and vertex set of a directed graph. IEEE Trans. Circuit Theory, 13(4): 399-403, 1966.

[45] B. Monjardet, Tournois et ordres médians pour une opinion, Math Sci. Humaines 43 (1973), 55-70.

[46] K.B.Reid, On serts of arcs containing no cycles in tournaments, Canad. Math Bulletin 12 (1969) 261-264.

[47] K.D.Reid and E.T.Parker, Disproof of a conjecture of Erdoe and Moser on tournaments, J. Combin. Theory 9 (1970) 225-238.

[48] R. Remage and W.A.Thompson, Maximum likelihood paired comparison rankings, Biometrika 53 (1966) 143-238.

[49] S. Seshu and M.B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, Reading, MA, 1961. Attribute the problem of the minimum feedback arc set to Runyon.

[50] P.D.Seymour. Packing directed circuit fractionally. Combinatorica 15:281-288, 1995.

[51] P. Slater, Inconsistencies in a schedule of paired comparisons, Biometrika 48 (1961) 303-312.

[52] J. Spencer, Optimal ranking of tournaments, Networks 1(1971) 135-138.

[53] J. Spencer, Optimal ranking of unrankable tournaments, Period. Math. Hungar. 11-2 (1980) 131-144.

[54] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. IEEE Trans. Syst. Man Cybern., 11(2):109-125, 1981.

[55] W.A.Thompson and R.Remage, Rankings from paired comparisons, Ann. Math. Statist. 35 (1964), 739-747.

[56] A.W.Tucker. On directed graphs and integer programs. Presented at 1960 Symposium on Combinatorial Problems, Princeton University, cited in [57].

[57] D.H. Younger. Minimum feedback arc sets for a directed graph. IEEE Trans. Circuit Theory 10 (1963), 238-245.

[58] A. van Zuylen. Deterministic approximation algorithms for ranking and clusterings. Cornell ORIE Tech report No. 1431. 2005. See also: Anke van Zuylen and Rajneesh Hegde and Kamal Jain and David P. Williamson Deterministic pivoting algorithms for constrained ranking and clustering problems. ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007, to appear.

# A Proofs and calculations

### Proof of lemma 4

*Proof.* By definition of rounding, for every $\rho$ we have:

$$|\tilde{C}(\rho) - C(\rho)| \leq \binom{n}{2} \epsilon b/(2n^2) \leq \epsilon b/4. \tag{3}$$

By definition of $\tilde{OPT}$, for every $\rho$ we have $\tilde{OPT} \leq \tilde{C}(\rho) \leq C(\rho) + \epsilon b/4$. Minimizing over $\rho$ gives

$$\tilde{OPT} \leq OPT + \epsilon b/4. \tag{4}$$

We first analyze the case where $OPT \geq b(1/2 - \epsilon)$. Using Equation 3, our assumption on the output ordering $\pi$, and Equation 4, we get

$$C(\pi) \leq \tilde{C}(\pi) + \epsilon b/4 \leq (1+\epsilon)\tilde{OPT} + \epsilon b/4 \leq (1+\epsilon)OPT + (1+\epsilon)\epsilon b/4 + \epsilon b/4.$$

Since $OPT \geq b(1/2 - \epsilon)$, this implies

$$C(\pi) \leq OPT \left[1 + \epsilon(1 + \frac{1+\epsilon/2}{1-2\epsilon})\right] \leq OPT(1+3\epsilon).$$

We next analyze the case where $OPT < b(1/2 - \epsilon)$. Let $\sigma$ be the cheapest ordering, such that $C(\sigma) = OPT$. By definition of $b$, for every pair $\{x, y\}$ we have $\max(w_{xy}, w_{yx}) \geq b/2 > OPT$, so $\sigma$ must be exactly the ordering such that $\sigma(x) < \sigma(y)$ iff $w_{xy} < w_{yx}$, and has cost exactly $\sum_{x,y} \min(w_{xy}, w_{yx})$, and for every pair we must have $\min(x, y) < b(1/2 - \epsilon)$. Consider any other ordering $\rho$: for every pair, $\rho$ costs at least as much as $\sigma$, and since $\rho \neq \sigma$, there must exist at least one pair $\{x, y\}$ where $\rho$ pays $\max(w_{xy}, w_{yx}) > \min(w_{xy}, w_{yx}) + \epsilon b$, and so $C(\rho) > C(\sigma) + \epsilon b$. From Equation 3, it follows that $\tilde{C}(\rho) > \tilde{C}(\sigma) + \epsilon b/2$. Since by Equation 4 we have $\tilde{OPT} = \tilde{C}(\sigma) < b/2$, this implies that $\tilde{C}(\rho) > (1+\epsilon)\tilde{OPT}$. But by assumption, the algorithm outputs a ordering $\pi$ such that $\tilde{C}(\pi) \leq (1+\epsilon)\tilde{OPT}$. So it has no choice but to output $\pi = \sigma$, the optimal ordering: the algorithm is optimal in this case. □

### Proof of Claim 5

*Proof.* By [19] Theorem 2 relating Spearman's footrule and Kendall-Tau, we have:

$$\begin{aligned}
\sum_j |\pi(j) - \sigma(j)| &\leq 2 \sum_{i,j} 1\!\!1\left(\sigma(i) > \sigma(j)\right) 1\!\!1\left(\pi(i) < \pi(j)\right) \\
&\leq \frac{2}{b} \sum_{i,j} 1\!\!1\left(\sigma(i) > \sigma(j)\right) w_{ij}^{\sigma} + 1\!\!1\left(\pi(i) < \pi(j)\right) w_{ij}^{\pi} \\
&= (2/b)(C(\sigma) + C(\pi)),
\end{aligned}$$

where the second inequality follows from the fact that if $\sigma(i) > \sigma(j)$ and $\pi(i) < \pi(j)$, then $\frac{1}{b}(w_{ij}^{\pi} + w_{ij}^{\sigma}) = \frac{1}{b}(w_{ij} + w_{ji}) \geq 1$.

□

**Proof of Claim 12**

$\psi_v^R$ is the same as $\psi_v$ but with the roles of $\pi$ and $\sigma$ reversed, so we need only prove the claim for $\psi_v$.

The following intermediate claim is useful:

**Claim 18.** $\sum_{1\le i\le |V|} \psi_v(i) \le F_v$.

*Proof.*

$$\psi_v(i) = \sum_{j\in v} \mathbb{1}\left(\pi(j) > i \text{ and } \sigma(j) \le i\right)$$

$$\sum_i \psi_v(i) = \sum_{j\in v}\sum_i \mathbb{1}\left(\sigma(j) \le i < \pi_j\right) = \sum_{j\in v:\pi(j)>\sigma(j)} (\pi(j)-\sigma(j)) \le \sum_{j\in v} |\pi(j)-\sigma(j)| = F_v$$

□

A second helper claim:

**Claim 19.** $\forall i, |\psi_v(i) - \psi_v(i+1)| \le 1$.

*Proof.*

$$\begin{aligned}
\psi_v(i) &= |\{j\in v|\sigma(j)\le i<\pi(j)\}| \\
&= |\{j\in v|\pi(j) > i+1 \text{ and } \sigma(j)\le i\}| + \mathbb{1}\left(\exists j\in v : \pi(j) = i+1 \text{ and } \sigma_j \le i\right) \\
\psi_v(i+1) &= |\{j\in v|\pi(j) > i+1 \text{ and } \sigma(j)\le i+1\}| \\
&= |\{j\in v|\pi(j) > i+1 \text{ and } \sigma(j)\le i\}| + \mathbb{1}\left(\exists j\in v : \pi(j) > i+1 \text{ and } \sigma(j) = i+1\right) \\
\psi_v(i+1)-\psi_v(i) &= \mathbb{1}\left(\exists j\in v : \pi(j) > i+1 \text{ and } \sigma(j) = i+1\right) - \mathbb{1}\left(\exists j\in v : \pi(j) = i+1 \text{ and } \sigma_j \le i\right)
\end{aligned}$$

Both terms are either 0 or 1, so the difference is in $[-1,1]$.

□

Now proof of claim 12:

*Proof.*

$$\psi_v(n) = |\{j\in v|\pi(j) > n \text{ and } \sigma(j)\le n\}| = 0 \text{ and } \psi_v(0) = |\{j\in v|\pi(j) > 0 \text{ and } \sigma(j)\le 0\}| = 0$$

Suppose, for a contradiction, that there exists $k$ such that $\psi_v(k) > \sqrt{F_v}$. Using Claim 19,

$$\sum \psi_v(i) \ge \psi_v(k) + 2\sum_{i=1}^{\psi_v(k)-1} i = \frac{\psi_v(k)(\psi_v(k)+1) + (\psi_v(k)-1)\psi_v(k)}{2} = \psi_v(k)^2 > F_v.$$

This contradicts Claim 18.

□

**Proof of Claim 13.**

*Proof.* Let $f(F_1, s_1) = F_1^{3/2}/s_1 + (F - F_1)^{3/2}/(s - s_1)$. Differentiating yields a unique stationary point of $F_1 = F/2$, $s_1 = s/2$, with value $\sqrt{2}F^{3/2}/s$. The minimum point along the boundary occurs when $s_1 = s/3$ and $F_1 = F/5$ with value $\sqrt{9/5}$. This is smaller than the stationary point in the interior, so this is the global minimum over this set. Therefore $\sqrt{9/5}F^{3/2}/s \leq f(F_1, s_1)$, so $F^{3/2}/s \leq \sqrt{5/9}\left[F_1^{3/2}/s_1 + (F - F_1)^{3/2}/(s - s_1)\right]$. □