# Fixing a Tournament

**Virginia Vassilevska Williams**
Computer Science Division
UC Berkeley
Berkeley, California 94720

## Abstract

We consider a very natural problem concerned with game manipulation. Let $G$ be a directed graph where the nodes represent players of a game, and an edge from $u$ to $v$ means that $u$ can beat $v$ in the game. (If an edge $(u, v)$ is not present, one cannot match $u$ and $v$.) Given G and a "favorite" node $\mathcal{A}$, is it possible to set up the bracket of a balanced single-elimination tournament so that $\mathcal{A}$ is guaranteed to win, if matches occur as predicted by $G$? We show that the problem is NP-complete for general graphs. For the case when $G$ is a tournament graph we give several interesting conditions on the desired winner $\mathcal{A}$ for which there exists a balanced single-elimination tournament which $\mathcal{A}$ wins, and it can be found in polynomial time.

## Introduction

Many championships use a single-elimination (also called knockout) format: the tournament proceeds in rounds; in each round players are paired up to play a game; the round winners move on to the next round, whereas the losers leave the tournament. This format is very common in sports. It also appears in the area of voting protocols where it is studied as the (binary) cup voting rule (Chevaleyre et al. 2007).

How far a particular player can go in a single-elimination tournament can vary vastly depending on the initial tournament bracket set-up. This work investigates the extent to which a tournament designer can influence the final tournament outcome by manipulating the initial brackets. We focus on the following set-up. Suppose we are to design a single-elimination tournament for some set of players. For each pair of players we have some information (obtained from history or by some other means) about which player is more likely to win in a match-up between the two. We have a favorite player $\mathcal{A}$ in mind, and we want to set the bracket for the tournament so that $\mathcal{A}$ has a very high chance of winning. We call this problem *tournament fixing*.

In this paper we consider two versions of the tournament fixing problem. In both versions we assume knowledge of the exact outcome of all match-ups between players.

1. In the first version only certain matches between players are allowed.
2. In the second version we can match any two players against each other. This is a well known formulation of the problem also known as *agenda control* (Bartholdi, Tovey, and Trick 1992).

In the first, more general version we show that tournament fixing is NP-hard; *i.e.*, if P$\neq$NP even if we have perfect information about the match outcomes, if some players cannot be matched with each other, it is infeasible to find a winning bracket layout for a given player.

This problem version is a special case of the following problem: we are given complete match outcome information and in addition, for every pair of players we are given an integer weight, corresponding to the revenue that would be generated if the two players are matched in the tournament. The problem is to find a single-elimination tournament which maximizes the total tournament revenue. If the weights capture "interestingness", the problem is to find the most interesting single-elimination tournament. This revenue maximization problem has been shown by (Lang et al. 2007) to be NP-hard, and the NP-hardness of general tournament fixing provides an alternative proof of this.

The second version of the problem allows us to relate single-elimination tournaments to *round-robin* tournaments: tournaments in which every pair of players has played and we know the outcome for each pair. Round-robin tournaments are widely studied. A significant body of work concerns the problem of optimally ranking players: finding a linear order of the players, a *ranking*, which minimizes the number of pairs $(u, v)$ such that $u$ beat $v$ but $v$ is before $u$ in the ranking. This objective dates back to Slater (Slater 1961) and has only recently been proven NP-hard (Alon 2006; Charbit, Thomassé, and Yeo 2007). Nevertheless, it has been shown (Coppersmith, Fleischer, and Rudra 2006) that the simple heuristic of ranking the players by the number of matches they have won approximates the optimal ranking within a factor of 5.

Our study uncovers interesting relationships between round-robin and single-elimination tournaments. For instance, we show that any player that beats a maximal number of players in a round-robin tournament (and is hence highly ranked), can win a single-elimination tournament, given the same match outcomes. We focus on the special case of the tournament fixing problem in which the favorite node is a *king*: for any player $b$ that it cannot beat, there is some player that it can beat that beats $b$. We give several sufficient conditions for which a king is a single-elimination tournament winner. One of our results is for the case when $\mathcal{A}$ is a very strong king: for any player $b$ that $\mathcal{A}$ cannot beat, there are at least $\log n$ players that $\mathcal{A}$ can beat who beat $b$. We show that such *super-kings* can win a single-elimination tournament, even though they may only be able to beat very few players head-to-head. An interesting consequence of this is that any tournament graph generated using the noisy sampling model of (Braverman and Mossel 2008) with error rate as low as $\Omega(\sqrt{\frac{\log n}{n}})$ has all its players as potential single-elimination tournament winners, with high probability. All of our arguments are constructive and allow us to design tournament brackets efficiently.

**Prior and Related Work.** There is considerable prior research on manipulating the outcome of a single-elimination tournament. Typically, the input to the studied problem is a tournament graph with probabilities on the edges. One then seeks to find a single-elimination tournament which maximizes the probability that a particular node will win. A major focus in such research is to maximize the winning probability of the *best* player under some assumptions (e.g., (Appleton 1995; Horen and Riezman 1985)). A common assumption is that the probability of a player beating another depends on the intrinsic abilities of the players which are hidden values. Moreover, the probability is monotone in the sense that every player has a higher probability of beating a weaker player than a stronger one. With these assumptions some positive results are possible, especially for small tournaments (Appleton 1995; Horen and Riezman 1985). On the other hand, if the given tournament graph has arbitrary probabilities and one wishes to determine whether a given node can win a single-elimination tournament with high probability, then the known results are mostly negative: this general problem is known to be NP-hard (Lang et al. 2007; Hazon et al. 2008). Even in the case when the probabilities are in $\{0, 1, 1/2\}$, the problem is NP-hard, as shown by Vu et al. (Vu, Altman, and Shoham 2008; 2009). The major open problem is whether tournament fixing for unweighted tournament graphs is NP-hard.

Tournament manipulation is also studied under the context of voting (Brams and Fishburn 2002; Hemaspaandra, Hemaspaandra, and Rothe 2007; Lang et al. 2007). In this context, the candidates are competing in an election based on majority comparisons along a binary voting tree. In each comparison, the candidate with fewer votes is eliminated, and the winner moves on to the next comparison. This setting is essentially a single-elimination tournament, the result of each match of which is known in advance: the corresponding tournament graph is unweighted. It is known that if the voting tree has no prescribed structure, then there is a polynomial time algorithm to decide whether there exists a voting tree for which a given candidate wins the election. Finding a *balanced* voting tree in polynomial time is a major open problem and is known in this area as the *agenda control* problem (Bartholdi, Tovey, and Trick 1992). (Fischer, Procaccia, and Samorodnitsky 2009) consider whether (potentially unbalanced) voting trees can be used to always elect a candidate preferred by close to the majority of voters; *i.e.* how well binary tree protocols approximate the Copeland winner. Fischer et al. give hardness results for deterministic trees, and show that by randomizing over voting trees one can obtain much better approximations.

**Preliminaries** For any graph $G = (V, E)$, $n = |V|$ and $m = |E|$, unless otherwise noted. For any node $v \in V$, let $N_{in}(v) = \{x \in V \mid (x, v) \in E\}$ and $N_{out}(v) = \{x \in V \mid (v, x) \in E\}$. If $v \in V$ and $S \subseteq V$, let $N_{in,S}(v) = \{x \in S \mid (x, v) \in E\}$ and $N_{out,S}(v) = \{x \in S \mid (v, x) \in E\}$. The *length* of a path in a graph is the number of edges on the path. An *arborescence* is a rooted tree such that all edges are directed away from the root. The *height* of an arborescence is the length of the longest path from the root to a leaf. For integer $n \geq 1$, let $[n] = \{1, \ldots, n\}$. A *tournament graph* is a directed graph $G = (V, E)$ such that for every pair $u, v \in V$, exactly one of $(u, v)$ or $(v, u)$ is in $E$. A node $a$ in a directed graph $G = (V, E)$ is a *king* if for any node $b \in V \setminus \{a\}$, either $(a, b) \in E$, or there exists a node $c \in V$ with $(a, c) \in E$ and $(c, b) \in E$.

A *binomial* arborescence $T = (V(T), E(T))$ rooted at $a \in V(T)$ is defined recursively as follows:

- a single node $a$ is a binomial arborescence rooted at $a$;
- if $|V(T)| = 2^i$ for some $i > 0$, then $T$ is a binomial arborescence if $a$ has a child $b$, (*i.e.*, $(a, b) \in E(T)$) such that if $T_b$ is the subarborescence of $T$ rooted at $b$ and $T_a = T \setminus T_b$, then $T_a$ and $T_b$ are $|V(T)|/2 = 2^{i-1}$–node binomial arborescences rooted at $a$ and $b$ respectively.

$T = (V(T), E(T))$ is a binomial *spanning* arborescence for a graph $G = (V, E)$ if $V(T) = V, E(T) \subseteq E$ and $T$ is a binomial arborescence.

## Binomial Arborescences and Hardness

Tournament graphs are widely used in the study of round-robin tournaments. The nodes of a tournament graph typically represent players, and an edge from $u$ to $v$ in the graph states that $u$ beats $v$. We use the same representation. However, since in our more general formulation of tournament fixing some match outcomes are not needed, we do not need an edge between each pair of vertices. Our abstraction of the *tournament fixing* problem (TFP) can be formalized as follows: Let $G = (V, E)$ be a directed graph on $n$ nodes such that if $(u, v) \in E$ then $(v, u) \notin E$. We are given a node $\mathcal{A} \in V$ and we are asked whether in $G$ there exists a spanning binomial arborescence rooted at $\mathcal{A}$. If this is the case, we call $\mathcal{A}$ a *binomial winner* in $G$.

Binomial arborescences on $n$ nodes represent single-elimination tournaments in the following sense: the node at which the arborescence is rooted is the winner of the tournament and its $\log n$ children are the players the root beats in the $\log n$ rounds; for $i = 1, \ldots, \log n$ the $i$th child of the root has $n/2^{\log n - i + 1}$ descendents, and the subarborescence rooted at that child inductively represents the subtournament which that child had to win to get to round $i$ in which it lost to the root player. The edge between the root and the $\log n$-th child represents the final of the tournament.

TFP is a spanning arborescence isomorphism problem for directed graphs. A spanning tree isomorphism problem in an *undirected* graph has the following form: given a fixed property $P$ of trees checkable in polynomial time and a graph $G$, find a spanning tree of $G$ satisfying $P$. (Papadimitriou and Yannakakis 1982) gave necessary and sufficient conditions under which a spanning tree isomorphism problem in an *undirected* graph is NP-complete. Their result implies that finding a binomial spanning tree in an undirected graph is NP-hard. The Papadimitriou and Yannakakis construction does not immediately imply that TFP is NP-complete, as TFP is a problem on *directed* graphs. It is possible that their proofs can be modified to work for rooted arborescences, although we do not know of such a result for directed graphs. We include a simple NP-completeness proof for TFP below.

**Theorem 1.** *TFP is* NP*-complete.*

*Proof.* TFP is clearly in NP. To show NP-hardness, we give a reduction from Exact Cover by 4-Sets which is well known to be NP-complete (Karp 1972; Garey and Johnson 1979). A pictorial representation of the reduction appears in Figure 1. Let $(U, \{S_1, \dots, S_k\})$ be an instance of Exact Cover by 4-sets, where $|U| = 4n$ and $|S_i| = 4$ for all $i \in [k]$. The problem is to determine whether there is a way to pick $n$ sets from $S$ which are disjoint and cover all elements of $U$. W.l.o.g. $k$ is a power of 2: we can always add copies of the same set $S_i$ to the instance. We will construct an instance of TFP: a digraph $G$ on $16k$ nodes, and a node $x_1$ of $G$ for which one would need to fix the tournament.

First, create a node $u$ for each $u \in U$, slightly abusing notation. For every set $S_i = \{u_1, u_2, u_3, u_4\}$ create 4 nodes, $s_1^i, \dots, s_4^i$. Add edges $(s_1^i, s_2^i), (s_1^i, s_3^i), (s_2^i, s_4^i)$. This creates a binomial arborescence $\bar{S}_i$ of size 4 rooted at $s_1^i$. Further, add edges $(s_2^i, u_1)$, $(s_3^i, u_2)$, $(s_4^i, u_3)$, $s_1^i, u_4)$. This makes $(s_1^i, s_2^i, s_3^i, s_4^i, u_1, u_2, u_3, u_4)$ a binomial arborescence of size 8 rooted at $s_1^i$.

Create $k - n$ disjoint binomial arborescences of size 4, $T_1, \dots, T_{k-n}$, rooted at $v_1, \dots, v_{k-n}$ respectively. Add edges $(v_i, s_1^j)$ for all $i = 1, \dots, k - n$ and $j = 1, \dots, k$. Then create $k$ disjoint binomial arborescences of size 8, $X_1, \dots, X_k$, with roots $x_1, \dots, x_k$, respectively. Add edges $(x_i, v_i)$ for $i = 1, \dots, k - n$, and $(x_j, s_1^t)$ for $j = (k - n + 1), \dots, k$ and $t = 1, \dots, k$. Finally, create a binomial arborescence $B_k$ on $\{x_1, \dots, x_k\}$ rooted at $x_1$. This completes the construction of graph $G$. The number of nodes in $G$ is $4n + 4k + 4(k - n) + 8k = 16k$, a power of 2.

The construction ensures that for any binomial arborescence $B$ spanning $G$:

- for $i = 1, \dots, k$, $\bar{S}_i$ is a subarborescence of $B$,
- for $i = 1, \dots, k - n$, $T_i$ is a subarborescence of $B$,
- for $i = 1, \dots, k$, $X_i$ is a subarborescence of $B$,
- $B_k$ is a subarborescence of $B$, and $B$ is rooted at $x_1$,
- in $B$, the elements of $U$ are partitioned into groups of 4 each of which is attached to some $\bar{S}_i$ forming a binomial arborescence of size 8,
- in $B$, each subarborescence $\bar{S}_i$ which does not have elements of $U$ attached to it, is linked to some $T_j$ to form a binomial arborescence of size 8,

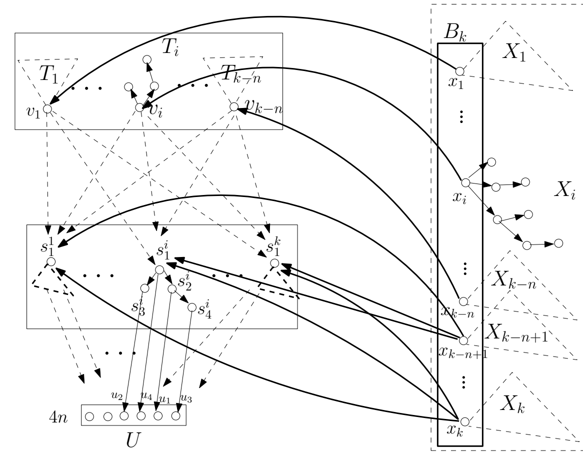


Figure 1: This is a pictorial description of the reduction from Exact Cover by 4-Sets to TFP.

- each $T_j$ is linked to some arborescence $\bar{S}_i$ to form a binomial arborescence of size 8,
- each $X_j$ with $j \leq k-n$ is linked to $T_j$ forming a binomial arborescence of size 16 rooted at $x_j$,
- each $X_j$ with $j > k - n$ is linked to some arborescence $\bar{S}_i$ which itself is linked to 4 elements of $U$, so that this forms a binomial arborescence of size 16 rooted at $x_j$.

The above points state that if there is a binomial spanning arborescence $B$ in $G$ rooted at $x_1$, then there is an exact cover of $U$ using the sets $S_i$ for which the arborescences $(s_1^i, s_2^i, s_3^i, s_4^i)$ are linked to 4 elements of $U$ in $B$. Vice versa, if there is an exact cover $C = \{S_{i_1}, \dots, S_{i_n}\}$, we can create a binomial arborescence as follows: for each $S_{i_j} = \{u_1, u_2, u_3, u_4\}$, attach the 4 nodes $u_1, u_2, u_3, u_4$ to $\bar{S}_{i_j}$. Match the $k - n$ sets $S_i \notin C$ to distinct arborescences $T_j$ and attach $\bar{S}_i$ to its corresponding $T_i$. We have a collection of $k$ size 8 binomial arborescences. We can attach each of them to some $X_t$ to form $k$ binomial arborescences of size 16. By adding the edges of $B_k$ we obtain a full size $16k$ binomial arborescence rooted at $x_1$. $\square$

It is unclear whether it is possible to modify this proof to show that TFP is NP-complete for tournament graphs. However, we can modify it to show such a result for complete *binary* arborescences (the proof appears in the full version of the paper). Binomial arborescences are similar to binary arborescences in that their height is logarithmic; perhaps this is an indication that tournament-TFP is NP-complete. However, binomial spanning arborescences are also similar to Hamiltonian paths, in that both structures always exist in a tournament graph. A Hamiltonian path rooted at a given node in a tournament can be found in polynomial time; perhaps this is an indication that tournament-TFP is in P.

**Theorem 2.** *Given a tournament graph $G$ and a node $\mathcal{A}$, deciding whether there is a complete binary arborescence spanning $G$ rooted at $\mathcal{A}$ is* NP*-complete.*

## When All Matches Are Allowed

In this section we consider the problem of fixing a tournament when we are allowed to match any pair of players. In the graph representation we are given a tournament graph and a node $\mathcal{A}$ and we need to find a binomial spanning arborescence rooted at $\mathcal{A}$. Here we consider some conditions under which a king in a tournament graph is a binomial winner. We focus on kings because they are strong players in the sense that they either have a very high winning record, or can beat some very highly ranked players. Moreover, for any graph $G$, if $G$ contains a binomial spanning arborescence rooted at a node $\mathcal{A}$, then any breadth first search arborescence starting from $\mathcal{A}$ must span $G$ and must have height at most $\log n$. The simplest nontrivial case for which this necessary condition is satisfied is when $\mathcal{A}$ is a king.

Tournament graphs are typically used to represent the outcomes of round-robin tournaments, and the nodes/players are often ranked by their outdegree, also called score. As mentioned in the introduction, the outdegree ranking is a good approximation to the optimal tournament ranking in the sense that it minimizes (within a constant factor) the number of matches in which a lower ranked player beat a higher ranked player. A node which has maximal outdegree, *i.e.*, outdegree at least as high as that of any other vertex, is among the top players in the round-robin ranking. We show that we can design a single-elimination tournament for any maximal degree node so that it would win given the same match outcomes. This gives an interesting and intuitive relation between round-robin and single-elimination tournaments – a very strong player should be able to win either. We give proofs of two more general statements which both imply the result for maximal degree nodes.

When a node has maximal outdegree, it is also a king in the tournament graph (this is the usual proof that a king always exists). Yet a node $\mathcal{A}$ is also a king if it does not necessarily have maximal outdegree, but for every node $b$ that $\mathcal{A}$ cannot beat, the outdegree of $\mathcal{A}$ is at least as large as that of $b$. The first statement we prove states that any such node $\mathcal{A}$ is a binomial winner. We note that the above outdegree condition allows for the outdegree of $\mathcal{A}$ to be as low as $n/3$. In contrast, a node with maximal outdegree has outdegree at least $n/2$.

In the second part of this section we show that any king with outdegree at least $n/2$ is a binomial winner. Furthermore, we show that this condition is tight in the sense that for any $n$ power of 2 there exists a tournament graph and a king $\mathcal{A}$ with outdegree $n/2 - 1$ such that $\mathcal{A}$ cannot win any single-elimination tournament. Finally, we turn our attention to very strong kings, called super-kings, and show that such nodes are binomial winners even if they have low outdegree (as low as $\log n$).

**Nodes Stronger than the Nodes that Beat Them.**

**Theorem 3.** *Let $G = (V, E)$ be a tournament graph. Let $\mathcal{A} \in V$ such that for all $v \in N_{in}(\mathcal{A})$, $deg_{out}(v) \leq deg_{out}(\mathcal{A})$. Then one can construct in polynomial time a binomial spanning arborescence of $G$ rooted at $\mathcal{A}$.*

*Proof.* Suppose we have partitioned the vertices of $B$ into binomial arborescences (of possibly different sizes) rooted at nodes of $A$. Lemma 1 below allows us to do that. Consider the vertices of $A$ which were not used in creating these arborescences by the Lemma. Partition these vertices (arbitrarily) into sets of sizes powers of 2, creating corresponding binomial arborescences (arbitrarily). Doing this, we partition $A \cup B$ into disjoint binomial arborescences $\{S_1, S_2, \ldots\}$ rooted at nodes of $A$, each of a size power of 2. Now, if there are two arborescences of the same size $2^k$, link them by adding the edge between their roots to create a binomial arborescence of size $2^{k+1}$. Continue doing this until there is at most one arborescence of each size. Because $|A \cup B| = 2^{\log n} - 1 = \sum_{i=0}^{\log n - 1} 2^i$, there will be a binomial arborescence of size $2^i$ for every $i$ from 0 to $\log n - 1$. Link the root of each of these arborescences to $\mathcal{A}$. Since the arborescences are rooted at vertices from $A$ this will form a binomial arborescence of size $n$ rooted at $\mathcal{A}$. $\square$

Now it remains to prove Lemma 1. We begin by showing the claim below.

**Claim 1.** *Let $A = N_{out}(\mathcal{A})$ and $B = N_{in}(\mathcal{A})$ in a tournament graph. Suppose for all $b \in B$, $deg_{out}(b) \leq deg_{out}(\mathcal{A})$. Then $\forall b \in B$, $deg_{out,B}(b) < deg_{in,A}(b)$.*

*Proof of Claim 1:* For any $b \in B$, $deg_{out}(b) \leq deg_{out}(\mathcal{A})$. Also, $deg_{out}(b) = deg_{out,B}(b) + deg_{out,A}(b) + 1$, $deg_{out}(\mathcal{A}) = |A|$, and $deg_{out,A}(b) = |A| - deg_{in,A}(b)$. Hence, $1 + deg_{out,B}(b) + |A| - deg_{in,A}(b) \leq |A|$, and $deg_{out,B}(b) < deg_{in,A}(b)$. $\square$

**Lemma 1.** *Given nonempty sets $A' \subseteq A$ and $B' \subseteq B$ such that for all $b \in B'$, $deg_{out,B'}(b) < deg_{in,A'}(b)$, one can pick a node $a' \in A'$ and a subset $S \subseteq N_{out,B'}(a')$ so that*

1. *$|S \cup \{a'\}| = 2^k$ for some integer $k \geq 1$, and*
2. *$\forall b' \in B' \setminus S$, $deg_{out,B' \setminus S}(b') < deg_{in,A' \setminus \{a'\}}(b')$.*

*Proof.* Since $B'$ is nonempty and for all $b \in B'$, $deg_{out,B'}(b) < deg_{in,A'}(b)$, there exists some $a' \in A'$ which has an out-neighbor in $B'$. Pick one such $a'$ and let $N = N_{out,B'}(a')$. For some integers $k \geq 1$ and $r$ we have $0 \leq r \leq 2^k - 1$ and $|N| = 2^k - 1 + r \geq 2r$. We can find a matching in $N$ of size at least $r$. Pick a submatching of size $r$, consisting of a set $R$ of $r$ vertices uniquely pointing to some other $r$ vertices in $N$. Let $S = N \setminus R$. Clearly, $|S| = 2^k - 1 + r - r = 2^k - 1$, and hence $|S \cup \{a'\}| = 2^k$. We will show that for all $b' \in B' \setminus S$, $deg_{out,B' \setminus S}(b') < deg_{in,A' \setminus \{a'\}}(b')$. Let $b' \in B' \setminus S$. If $b' \notin N$, then $deg_{out,B' \setminus S}(b') < deg_{in,A' \setminus \{a'\}}(b')$ since $deg_{out,B' \setminus S}(b') \leq deg_{out,B'}(b') < deg_{in,A'}(b')$. If $b' \in N$, then $deg_{in,A' \setminus \{a'\}}(b') = deg_{in,A'}(b') - 1$. Yet, since $b' \in N \cap \{B' \setminus S\}$, we must have $b' \in R$, and $b'$ must have at least one outneighbor in $S$. Hence $deg_{out,B' \setminus S}(b') \leq deg_{out,B'}(b') - 1$, and since $deg_{out,B'}(b') < deg_{in,A'}(b')$, $deg_{out,B' \setminus S}(b') < deg_{in,A' \setminus \{a'\}}(b')$. $\square$

**Kings who Beat Half the Players.** Suppose now that we have a king who is not necessarily the strongest player but is still relatively strong – he can beat at least half of the players. It is not immediately clear that this king is binomial winner. In fact, if the king can only beat $\frac{n}{2} - 1$ players, there are tournament graphs for which such a king may not be able to win at all:

**Claim 2.** *For any $n$, power of $2$, there exists a tournament graph on $n$ nodes with a king $\mathcal{A}$ with outdegree $n/2-1$ such that there is no binomial spanning arborescence rooted at $\mathcal{A}$.*

*Proof.* Consider the following graph $G$ and king node $\mathcal{A}$: let $A$ be the out-neighborhood of $\mathcal{A}$ and let $B$ be the in-neighborhood of $A$, so that $|A| = \frac{n}{2} - 1$ and $|B| = \frac{n}{2}$. Let $a' \in A$ be a node so that $a'$ beats every node in $B$. This makes $\mathcal{A}$ a king. If $x \in A \setminus \{a'\}$, then every node $b \in B$ beats $x$. The edges within $A$ and $B$ are arbitrary.

Then the only way for a binomial spanning arborescence $T$ to be rooted at $\mathcal{A}$ is if all nodes of $B$ are in the subarborescence rooted at $a'$: an element $b \in B$ can only be beaten by another element of $B$, or by $a'$. However, $|B \cup \{a'\}| > n/2$ and hence the height of the subarborescence of $T$ rooted at $a'$ is at least $\log n$. This means that this subarborescence is the entire $T$ and $T$ cannot be rooted at $\mathcal{A}$. $\square$

We now show that if a king can beat at least half of the players then he is a binomial winner. This result is tight by Claim 2.

**Theorem 4.** *Let $\mathcal{A}$ be a king in an $n$-node tournament graph $G$ so that $outdeg(\mathcal{A}) \geq n/2$. Then $\mathcal{A}$ is a binomial winner, and a binomial spanning arborescence rooted at $\mathcal{A}$ can be found in polynomial time.*

*Proof.* The proof will proceed by induction on $n$. The base case is $n = 2$ and then it is trivially true. We will keep the invariant $I$ that if $N$ players are left in the tournament, $\mathcal{A}$ is one of these players and $\mathcal{A}$ is a king with outdegree at least $N/2$ in the induced tournament graph. The induction hypothesis for $n$ is that if $I$ holds for a node $\mathcal{A}$ in a tournament graph $G'$ on $n/2$ nodes, then $\mathcal{A}$ is a binomial winner in $G'$. The proof will proceed by fixing a match-up for the first round of the tournament and showing that $I$ holds for the first round winners and $\mathcal{A}$.

Let $A = N_{out}(\mathcal{A})$ and $B = N_{in}(\mathcal{A})$. If $B$ is empty, we are done: $\mathcal{A}$ will win any tournament. Otherwise, in Round 1 create a maximal matching $\bar{M}$ from $A$ to $B$, and let $K$ be all elements of $A$ in $\bar{M}$; $k = |\bar{M}| = |K| \leq |B|$. The matching $\bar{M}$ ensures that all elements of $K$ survive round 1.

For the rest of the $|B| - k$ elements of $B$, pick any maximum matching $M$ of them; $|M| = \lfloor \frac{|B|-k}{2} \rfloor$. We will show that $A \setminus K$ is nonempty and we can pick an element $a'$ from it and match it with $\mathcal{A}$, so that $\mathcal{A}$ survives round 1. For this it suffices to show that $|A| - k \geq 1$.

If $|B| - k$ is odd, there is one element $b'$ of $B$ which remains unmatched. We will show that $A \setminus (K \cup \{a'\})$ is nonempty if $|B| - k$ is odd, and hence there is at least one element $a''$ which we can match with $b'$. For this it suffices to show that $|A| - k - 1 \geq 1$ if $|B| - k$ is odd.

There is an even number of remaining unmatched elements of $A$ as $n$ is even. Pick any matching on them. This completes round 1. Note that $\mathcal{A}$ and all elements of $K$ are winners in this round, and hence $\mathcal{A}$ remains a king. We must show two things:

1. $|A| - k \geq 1$, and if $|B| - k$ is odd, $|A| - k - 1 \geq 1$,
2. the number of elements in $A$ which survive round 1 is at least $n/4$.

First, $|A| + |B| = n - 1$ is odd, and hence $|B|$ is even iff $|A|$ is odd. Hence, $|B| - k$ is odd iff $|A| - k - 1$ is odd. Moreover, $k \leq |B| \leq |A| - 1$ as $|A| \geq n/2$, and hence $|A| - k - 1 \geq 0$ and $|A| - k \geq 1$. If $|B| - k$ is odd, then $|A| - k - 1 \geq 1$ since $|A| - k - 1$ is odd. We are done with part 1 above.

The number of elements in $A$ which survive round 1 is at least $k + \frac{|A|-k-1}{2}$ if $|B| - k$ is even and $k + \frac{|A|-k-2}{2}$ if $|B| - k$ is odd. This number is always $\lfloor \frac{|A|+k-1}{2} \rfloor$. Since $k \geq 1$, at least $\lfloor \frac{|A|}{2} \rfloor \geq \lfloor \frac{n}{4} \rfloor = \frac{n}{4}$ elements of $A$ survive round 1. After this round we have a tournament on $n/2$ elements in which $\mathcal{A}$ is a king with outdegree at least $n/4$. By induction, $\mathcal{A}$ is a binomial winner. $\square$

**Super-Kings.** We now define a *super-king* in a tournament graph as a node $\mathcal{A}$ with the constraint that for any $b \in N_{in}(\mathcal{A})$, $|N_{in,N_{out}(\mathcal{A})}(b)| \geq \log n$.

**Theorem 5.** *Let $G$ be a tournament graph and let $\mathcal{A}$ be a super-king in $G$. Then $\mathcal{A}$ is a binomial winner in $G$.*

*Proof.* Let $A = N_{out}(\mathcal{A})$ and let $B = N_{in}(\mathcal{A})$. We will proceed by induction. If $n = 1$ we are done. Otherwise if $n \geq 2$, notice that $|A| \geq \log n \geq 1$. Pick any node $a' \in A$ and match it with $\mathcal{A}$. Now, create a maximum matching $M$ from $A \setminus \{a'\}$ to $B$. The number of remaining unmatched nodes is $n - 2 - 2|M|$, which is even, hence we can pick some perfect matching $M'$. Call the final matching $N = \{(\mathcal{A}, a')\} \cup M \cup M'$.

Consider the set $S$ of $n/2$ nodes which are sources in $N$. If $b \in S \cap B$, then $|N_{in,A}(b) \setminus S| \leq 1$. This is since if $a'' \in (N_{in,A}(b) \setminus S)$ and $a'' \neq a'$, then after the creation of $M$, $a''$ was unmatched. Furthermore, since $b \in S$, $b$ must have been unmatched after the creation of $M$. This is a contradiction to the maximality of $M$. Thus, every $b$ in the new inneighborhood of $\mathcal{A}$ has at least $\log n - 1 = \log(n/2)$ in-neighbors in $A$. The theorem follows by induction on $n$. $\square$

**The Braverman–Mossel Noisy Sampling Model.** The following model was proposed in (Braverman and Mossel 2008). We are given a parameter $q < 1/2$ and an ordered list of $n$ players, $v_1, \ldots, v_n$ which represents a sorted order of the players by their *intrinsic abilities*. The parameter $q$ represents a noise rate. A tournament graph $T_q$ is generated as follows: for every pair of nodes $v_i, v_j$ with $j > i$ independently at random one places an edge $(v_i, v_j)$ with probability $1 - q$, and with probability $q$ the reverse edge $(v_j, v_i)$. The smaller $q$ is, the closer $T_q$ is to a transitive tournament; the larger $q$ is, the closer $T_q$ is to a completely random tournament. Here we show that $q$ can be very small and still *every* player in $T_q$ is a binomial winner, with high probability. In other words, almost all tournaments generated in this model (for slightly large $q$) can be fixed for any player.[1]

**Theorem 6.** *Let $q > 4\sqrt{\frac{\log n}{n}}$. Then with high probability, all nodes in a random tournament $T_q$ generated using the Braverman–Mossel model are binomial winners.*[2]

*Proof.* Let $q = c \cdot \sqrt{\frac{\log n}{n-1}}$ for some $c > 4$. We show that with high probability all nodes in $T_q$ are super-kings. Let $V = \{v_1, \ldots, v_n\}$. Call a node $v$ "bad" if either

- $deg_{out}(v) \le (c \log n)/q$, or
- $deg_{out}(v) > (c \log n)/q$ and there exists some node $b \ne v$ such that $|N_{in,N_{out}(v)}(b)| < \log n$.

Let $U = V \setminus \{v\}$. Let $u_i$ be the $i$th node of $U$. Let $Z_i$ be an indicator variable which is 1 if $(v, u_i)$ is an edge and 0 otherwise. Then $Pr[Z_i = 1] \ge q$. Let $Z = \sum_i Z_i$; $Z = deg_{out}(v)$. Then $E[Z] = \sum_i E[Z_i] \ge q(n-1) = c\sqrt{(n-1)\log n}$. By a Chernoff bound:

$$Pr[Z \le (c \log n)/q] \le e^{-E[Z](1-(c\log n)/(qE[Z]))^2/2} \le$$
$$e^{-(c(1-1/c)^2/2)\sqrt{(n-1)\log n}} = 1/2^{\Omega(\sqrt{n \log n})}.$$

Now fix some $b$. We are interested in the probability that $b$ has $< \log n$ inneighbors from $N_{out}(v)$, given that $deg_{out}(v) > (c \log n)/q$. Consider the random variable $Y = deg_{in,N_{out}(v)}(b) = \sum_{u \in N_{out}(v)} Y_u$, where $Y_u$ is 1 iff $(u, b)$ is an edge. $Pr[Y_u = 1] \ge q$ and so $E[Y] = q\, deg_{out}(v) > c \log n$. Again by a Chernoff bound:

$$Pr[Y < \log n] \le Pr[Y < (1/c)E[Y]] \le$$
$$e^{-(1-1/c)^2/2E[Y]} \le 1/n^{c(1-1/c)^2/(2\ln 2)}.$$

Let $C = -2 + c(1-1/c)^2/(2 \ln 2)$. By a union bound, the probability that $v$ is bad is at most $1/2^{\Omega(\sqrt{n\log n})} + 1/n^{C+1}$. The probability that there exists a bad $v$ is at most

$$1/2^{\Omega(\sqrt{n \log n})} + 1/n^C \le O(1/n^C).$$

Hence with probability at least $1 - O(1/n^C)$ all nodes are super-kings. $C$ is at least $(c-4)/(2\ln 2)$, and so $C > 0$ for any $c > 4$. $\square$

**Acknowledgments.** The author would like to thank David Abraham, Ryan Williams, Guy Blelloch, Maverick Woo, Anupam Gupta, Noga Alon and Piotr Faliszewski for some helpful discussions, and the anonymous reviewers for their valuable feedback.

This material is based upon work supported by the National Science Foundation under Grant CCF-0832797 at Princeton University/IAS and Award #0937060 to the Computing Research Association for the CIFellows Project.

[1] This result crucially relies on knowing the match outcomes beforehand: for any tournament bracket that we pick before the coin flips, $v_n$ has only a $q^{\log n}$ chance of winning the tournament which is at most $1/\text{poly}(n)$ even when $q = \Theta(1)$.

[2] The statement of this theorem was suggested to the author by an anonymous reviewer.

## References

Alon, N. 2006. Ranking tournaments. *SIAM J. Discret. Math.* 20(1):137–142.

Appleton, D. R. 1995. May the best man win? *The Statistician* 44(4):529–538.

Bartholdi, J.; Tovey, C.; and Trick, M. 1992. How hard is it to control an election? *Mathematical and Computer Modeling* 16(8/9):27–40.

Brams, S. J., and Fishburn, P. C. 2002. Voting procedures. *K. J. Arrow, A. K. Sen, and K. Suzumura, editors, Handbook of Social Choice and Welfare*.

Braverman, M., and Mossel, E. 2008. Noisy sorting without resampling. In *Proc. SODA*, 268–276.

Charbit, P.; Thomassé, S.; and Yeo, A. 2007. The minimum feedback arc set problem is NP-hard for tournaments. *Combinatorics, Probability & Computing* 16(1):1–4.

Chevaleyre, Y.; Endriss, U.; Lang, J.; and Maudet, N. 2007. A short introduction to computational social choice. *SOFSEM 2007: Theory and Practice of Computer Science* 4362:51–69.

Coppersmith, D.; Fleischer, L.; and Rudra, A. 2006. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proc. SODA*, 776–782.

Fischer, F.; Procaccia, A.; and Samorodnitsky, A. 2009. A new perspective on implementation by voting trees. In *Proc. EC*, 31–40.

Garey, M., and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company.

Hazon, N.; Dunne, P.; Kraus, S.; and Wooldridge, M. 2008. How to rig elections and competitions. In *Proc. COMSOC*.

Hemaspaandra, E.; Hemaspaandra, L. A.; and Rothe, J. 2007. Anyone but him: The complexity of precluding an alternative. *Artif. Intell.* 171(5/6):255–285.

Horen, J., and Riezman, R. 1985. Comparing draws for single elimination tournaments. *Operations Research* 33(2):249–262.

Karp, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations, R.E. Miller and J.W. Thatcher (eds.), Plenum Press, New York*, 85–104.

Lang, J.; Pini, M. S.; Rossi, F.; Venable, K. B.; and Walsh, T. 2007. Winner determination in sequential majority voting. In *Proc. IJCAI*, 1372–1377.

Papadimitriou, C. H., and Yannakakis, M. 1982. The complexity of restricted spanning tree problems. *J. ACM* 29(2):285–309.

Slater, P. 1961. Inconsistencies in a schedule of paired comparisons. *Biometrika* 48(3/4):303–312.

Vu, T.; Altman, A.; and Shoham, Y. 2008. On the agenda control problem in knockout tournaments. In *Proc. COMSOC*.

Vu, T.; Altman, A.; and Shoham, Y. 2009. On the complexity of schedule control problems for knockout tournaments. In *Proc. AAMAS*, 225–232.