Rigging Tournament Brackets for Weaker Players

Isabelle Stanton and **Virginia Vassilevska Williams**
Computer Science Department
University of California, Berkeley
isabelle,virgi@eecs.berkeley.edu

Abstract

Consider the following problem in game manipulation. A tournament designer who has full knowledge of the match outcomes between any possible pair of players would like to create a bracket for a balanced single-elimination tournament so that their favorite player will win. Although this problem has been studied in the areas of voting and tournament manipulation, it is still unknown whether it can be solved in polynomial time. We focus on identifying several general cases for which the tournament can always be rigged efficiently so that the given player wins. We give constructive proofs that, under some natural assumptions, if a player is ranked among the top K players, then one can efficiently rig the tournament for the given player, even when K is as large as 19% of the players.

1 Introduction

As a natural way to select a leader, competition is at the heart of life. It is intriguing, both for its participants, and its spectators. Society is riddled with organized competitions called *tournaments* with well-defined rules to select a winner from a pool of candidate players. Sports tournaments such as the FIFA World Cup and Wimbledon are immensely popular and generate huge amounts of revenue. Elections are another important type of tournaments: a leading party is selected according to some rules using votes from the population.

Two of the most common tournament formats employed in both sports and voting are *round-robin* and *single-elimination*. In the former, every pair of players are matched up, and a player's score is how many matches they won. If some player has beaten everyone else, then they are the clear (Condorcet) winner. Otherwise, the winner is not well-defined. However, given the outcomes of a round-robin tournament, there are various methods of producing rankings of the players. The most common definition of the optimal ranking is that it minimizes the number of wins of a lower-ranked player over a higher-ranked player [Slater, 1961]. Although finding such a ranking for a round-robin tournament is NP-hard [Alon, 2006], sorting the players according to their number of wins is a good approximation to the optimum ranking [Coppersmith *et al.*, 2006].

Single-elimination (SE) tournaments are played as follows. First, a permutation of the players, called the *bracket* or *schedule* is given. According to the bracket, the first two players are matched up, then the second pair of players etc. The winners of the matches move on to the next round. The bracket for this round is obtained by pairing up the remaining players according to the original bracket. If the number of players is a power of 2, the tournament is balanced. Otherwise, it is unbalanced and some players advance to the next round without playing a match. In practice, these *byes* are usually granted in the first round. Although the winner of an SE tournament is always well-defined, the chance of a particular player winning the tournament can vary immensely depending on the bracket. Arguably, this gives the tournament organizer a lot of power. The study of how much control an organizer has over the outcome of a tournament is called *agenda control* [Bartholdi *et al.*, 1992].

The most studied agenda control problem for balanced SE tournaments is to find a bracket which maximizes the probability that a given player will win the tournament. The tournament organizer is given the probability that i will beat j for every pair of players i, j. A major focus is to maximize the winning probability of the *strongest* player under some assumptions[1] (e.g., [Appleton, 1995; Horen and Riezman, 1985; Vu and Shoham, 2010b; 2010a]). Without assumptions on the probabilities, the agenda control problem for an arbitrary given player is NP-hard [Lang *et al.*, 2007; Hazon *et al.*, 2008], even when the probabilities are in $\{0, 1, 1/2\}$ [Vu *et al.*, 2010]. Moreover, the maximum probability that a given player wins cannot be approximated within any constant factor unless P=NP [Vu *et al.*, 2010]. When the probabilities are all either 0 or 1, the agenda control problem, then called the tournament fixing problem (TFP), is not

[1] A common assumption is monotonicity: the probability of beating a weaker player is at least as high as that of beating a stronger one.

well understood. One of the interesting open problems in computational social choice is whether a tournament fixing bracket can be efficiently found. Several variants of the problem are NP-hard – when some pairs of players cannot be matched [Vassilevska Williams, 2010], when some players must appear in given rounds [Vu *et al.*, 2010], or when the most "interesting" tournament is to be computed [Lang *et al.*, 2007].

Besides its natural connection to tournament manipulation, TFP studies the relationship between round-robin and single-elimination tournaments. The decision version of TFP asks, given the results of a round-robin tournament and a player $\mathcal{A}$, is $\mathcal{A}$ also the winner of some SE tournament, given the same match outcomes? In the area of voting, suppose all votes are in, can we simulate a win for a particular candidate, using single-elimination rules (binary cup)? In this work, we investigate the following question: if we consider a round-robin tournament and a ranking produced from it by sorting the players according to their number of wins, how many of the top players can actually win some SE tournament, given the same match outcomes? What conditions on the round-robin tournament suffice so that one can efficiently rig the SE tournament outcome for many of the top players?

Prior work has shown several intuitive results. For instance, if $\mathcal{A}$ is any player with the maximum number of wins in a round-robin tournament, then one can efficiently construct a winning (balanced) SE bracket for $\mathcal{A}$ [Vassilevska Williams, 2010]. We extend and strengthen many of the prior results.

Contributions. Let Π be an ordering of the players in nonincreasing order of their number of wins in the given round-robin tournament. We consider conditions under which, for large K, the SE tournament can be fixed efficiently for *any* of the first K players in Π. We are interested in natural and not too restrictive conditions under which a constant fraction of the players can be made to win. If the first player p_1 in Π beats everyone else, then p_1 wins all SE tournaments. We show that if *any* player can beat p_1, then we can also fix the tournament for the second player p_2. We show that for large enough tournaments, if there is a matching onto the top $K-1$ players $\{p_1, \ldots, p_{K-1}\}$ in Π from the rest of the players, then we can efficiently find a bracket for which p_K wins, where K is as large as 19% of the players.

Graph representation. The outcome of a round-robin tournament has a natural graph representation as a *tournament graph*: a directed graph in which for every pair of nodes a, b, there is an edge either from a to b, or from b to a. The nodes of a tournament graph represent the players in a round-robin tournament, and an edge (a, b) represents a win of a over b.

Notation and Definitions. Unless noted otherwise, all graphs in the paper are tournament graphs over n vertices, where n is a power of 2, and all SE tournaments are balanced. In Table 1, we define the notation that will be used in the rest of this paper. For the definitions, let $\mathcal{A} \in V$ be any node, let $X, Y \subseteq V$ be such that $X \cap Y = \emptyset$.

Notation
$N^{out}(\mathcal{A}) = \{v \mid (\mathcal{A}, v) \in E\},\ N^{out}_X(\mathcal{A}) = N^{out}(\mathcal{A}) \cap X$
$N^{in}(\mathcal{A}) = \{v \mid (v, \mathcal{A}) \in E\},\ N^{in}_X(\mathcal{A}) = N^{in}(\mathcal{A}) \cap X$
$out(\mathcal{A}) = \lvert N^{out}(\mathcal{A})\rvert,\ out_X(\mathcal{A}) = \lvert N^{out}_X(\mathcal{A})\rvert$
$in(\mathcal{A}) = \lvert N^{in}(\mathcal{A})\rvert,\ in_X(\mathcal{A}) = \lvert N^{in}_X(\mathcal{A})\rvert$
$\mathcal{H}^{in}(\mathcal{A}) = \{v \mid v \in N^{in}(\mathcal{A}), out(v) > out(\mathcal{A})\}$
$\mathcal{H}^{out}(\mathcal{A}) = \{v \mid v \in N^{out}(\mathcal{A}), out(v) > out(\mathcal{A})\}$
$\mathcal{H}(\mathcal{A}) = \mathcal{H}^{in}(\mathcal{A}) \cup \mathcal{H}^{out}(\mathcal{A})$
$E(X, Y) = \{(u, v) \mid (u, v) \in E, u \in X, v \in Y\}$

Table 1: A summary of the notation used in this paper.

Consider a tournament graph $G = (V, E)$. We say that $\mathcal{A} \in V$ is a king over another node $x \in V$ if either $(\mathcal{A}, x) \in E$ or there exists $y \in V$ such that $(\mathcal{A}, y), (y, x) \in E$. A *king* in G is a node $\mathcal{A}$ which is a king over all $x \in V \setminus \{\mathcal{A}\}$. We say that set S covers a set T if for every $t \in T$ there is some $s \in S$ so that $(s, t) \in E$. Thus $N^{out}(\mathcal{A})$ covers the graph if and only if $\mathcal{A}$ is a king.

If one can efficiently construct a winning SE tournament bracket for a player $\mathcal{A}$, we say that $\mathcal{A}$ is an *SE winner*. We use the ranking Π formed by sorting the players in nondecreasing order of their outdegree.

We will construct SE tournaments as a series of matchings where each successive one will be over the winners of the previous one. A matching is defined as a set of pairs of vertices where each vertex appears in at most one pair. In our setting, these pairs are directed, so a matching from X to Y will consist only of edges that are directed from X to Y. If an edge is directed from x to y, then we refer to x as a *source*. Further, given a matching M from the sets X to Y, we will use the notation $X \setminus M$ to refer to the vertices in X that are not contained in the matching. A perfect matching from X to Y is one where every vertex of X is matched with a vertex of Y and $|X| = |Y|$. A perfect matching in a set S is a perfect matching from some $S' \subseteq S$ to $S \setminus S'$.

2 Motivation and Counterexamples

We will now discuss the motivation for our assumptions on the graph. We will look at some necessary and sufficient conditions for the top K players to win an SE tournament. We begin with an example.

Consider the transitive tournament graph G with nodes $v_1, \ldots, v_n$, where v_i beats all nodes v_j for $j > i$. Then v_1 is the winner of all SE tournaments on G. Now, take any perfect matching from $\{v_1, \ldots, v_{n/2}\}$ to $\{v_{n/2+1}, \ldots, v_n\}$ and reverse these edges to create a *back-matching*. This gives each node from the weaker half of G a win against some node from the stronger half. The new outdegree ranking only swaps $v_{n/2}$ and $v_{n/2+1}$, however now the top $n/2 - 1$ players are SE winners: each of these nodes still beats at least $n/2$ other players, and the back-edges of the matching also make each one also a king. Prior work showed that this condition is

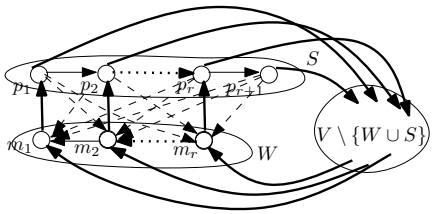

Figure 1: p_i only loses to m_i and p_j for $j < i$. No matter how the other edges of the tournament graph are placed, since the p_i beat everyone else and the m_i lose to everyone else, all SE tournament winners are in S.

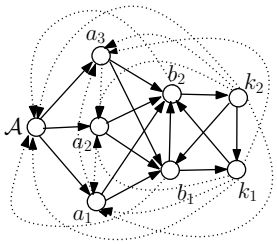

Figure 2: Example in which the two highest outdegree nodes, k_1 and k_2, have a matching into them but $\mathcal{A}$ cannot win an SE tournament.

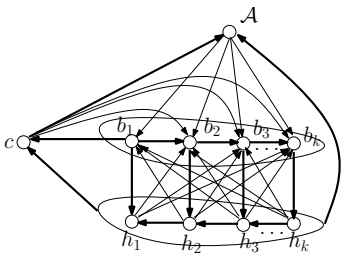

Figure 3: Example where there is a matching from $N^{out}(\mathcal{A})$ onto the k highest degree nodes but $\mathcal{A}$ can't win an SE tournament.

sufficient for these players to be an SE tournament winner [Vassilevska Williams, 2010]. Thus, adding a back-matching to a transitive tournament can dramatically increase the set of winners. Our goal is to understand the impact of such back-edge matchings in general tournaments. As a warm-up, we consider the nodes of second and third highest outdegree. By case analysis, one can show the following theorem.

Theorem 1. *Let G be a tournament graph and let $\mathcal{A}$ be the node of second highest outdegree. Then $\mathcal{A}$ is an SE winner if and only if there is no Condorcet winner in G. If there is a matching onto the top 2 nodes, then the third highest outdegree node is also an SE winner.*

This simple result leads to a larger question. What are the necessary and sufficient conditions for the k^{th} ranked node to win an SE tournament? A natural conjecture is that if there is a perfect matching from $V \setminus \mathcal{H}(\mathcal{A})$ to $\mathcal{H}(\mathcal{A})$, then $\mathcal{A}$ should be able to win.

In Figure 1 we give an example of a tournament and a subset S consisting of the top $r+1$ outdegree nodes such that there is a matching of size r from a subset $W = \{m_1, \ldots, m_r\}$ of $V \setminus S$ into S, but no matching of size $r+1$ from $V \setminus S$ into S. Figure 1 only shows some of the graph edges. The edges within $V \setminus (W \cup S)$ are arbitrary, all nodes of S beat all nodes of $V \setminus (W \cup S)$, and all nodes of W lose to all nodes of $V \setminus (W \cup S)$. We can show that any node $\mathcal{A} \notin S$ cannot be an SE winner. p_1 only loses to m_1 and m_1 loses to everyone else so p_1 must be matched with m_1 in the first round if it is to ever be eliminated. Similarly, for any $i \leq r$, each p_i must be matched with m_i in the first round. Since all of the nodes that could possibly beat p_{r+1} lose in the first round, no one is left to beat p_{r+1} and $\mathcal{A}$ cannot win. Therefore, the only possible SE winners are contained in S. We have shown that for any r there exists a graph in which there is no matching onto the top r outdegree nodes and the $(r+1)$st outdegree node is not an SE winner. From this, we can conclude that the existence of a perfect matching from $V \setminus \mathcal{H}(\mathcal{A})$ into $\mathcal{H}(\mathcal{A})$ is, in a sense, necessary, in order for a node $\mathcal{A}$ to be an SE winner.

Now suppose that there is a perfect matching in G from $V \setminus \mathcal{H}(\mathcal{A})$ onto $\mathcal{H}(\mathcal{A})$. Can we conclude that the bracket can be fixed for $\mathcal{A}$? This turns out not to be true. Consider Figure 2. Here $\mathcal{H}(\mathcal{A})$ consists only of k_1 and k_2. These nodes are only beaten by b_1 and b_2 respectively, who lose to every other player except $\mathcal{A}$, so b_i and k_i must be matched in round 1. The a_i are symmetric, so without loss of generality we can match $\mathcal{A}$ to a_1 in round 1. The two remaining nodes, a_2 and a_3, must also be matched. After round 1 the nodes that survive are $\mathcal{A}, a_3, b_1, b_2$. However, $\mathcal{A}$ needs to have outdegree at least 2 to survive the next two rounds. As it only has outdegree 1, $\mathcal{A}$ cannot win an SE tournament.

A similar problem can arise when the matching comes from $N^{out}(\mathcal{A})$ instead of $N^{in}(\mathcal{A})$. Figure 3 gives an example of a graph construction for any $n \geq 8$ for which the node ranked $n/2$ cannot win any SE tournament even though there is a matching onto $\mathcal{H}(\mathcal{A}) = \cup_{i=1}^{k} h_i$. Each h_i only loses to b_i and $\cup_{j>i} h_j$. Each b_i only beats $\cup_{j>i} b_j$, except for b_1 who also beats c. The problem arises with who to match $\mathcal{A}$ to in the first round so that it can win the match. By induction, one can argue that every h_i for $i > 1$ must be matched to b_i in round 1. $\mathcal{A}$ must be matched to some node in $N^{out}(\mathcal{A})$, but only b_1 remains unmatched. This leaves h_1 and c who must be matched as well. However, in round 2, all nodes that beat h_1 have been eliminated and it is now a Condorcet winner in the induced subgraph. Therefore, it must be the winner of any SE tournament.

A common issue in the above counterexamples is that $\mathcal{H}(\mathcal{A})$ is too large while $out(\mathcal{A})$ is too small. Another commonality is that $\mathcal{H}(\mathcal{A}) = \mathcal{H}^{in}(\mathcal{A})$. Hence a better condition to look for is a matching from $V \setminus \mathcal{H}^{in}(\mathcal{A})$ onto $\mathcal{H}^{in}(\mathcal{A})$, and not necessarily onto $\mathcal{H}(\mathcal{A})$.

Finally, a natural question is, how reasonable is the assumption of the existence of a matching from lower ranked players to higher ranked players? Consider the Braverman-Mossel model [2008] for generating tournament graphs. In this model, one assumes an underlying ranking $v_1 \cdots v_n$ of the players according to skill. The tournament is generated by adding an edge (v_i, v_j) with probability p if $j < i$ and $1-p$ if $j > i$ for $p < \frac{1}{2}$. This model can be viewed as a transitive tournament with each edge reversed with probability p. A classic result

of [Erdős and Rényi, 1964] is that a bipartite graph with n nodes on each side with $2n \ln n$ edges selected uniformly at random contains a perfect matching with high probability. If a graph is generated by the Braverman-Mossel model with $p > \frac{4 \ln n}{n}$, then we expect there to be $n \ln n$ back edges from $v_{n/2} \cdots v_n$ to $v_1 \cdots v_{n/2-1}$. Therefore, in almost all such tournaments, a backedge matching exists.

3 Main Results

We are now ready to introduce our main result. As the proof is quite technical, we will first provide an intuitive sketch, some of the necessary Lemmas, and a more detailed account of the key part of our proof. Please refer to [Stanton and Vassilevska Williams, 2011] for the full version of this Section.

We present two main results. The first generalizes the idea of a king, and shows that if a node $\mathcal{A}$ is a king except for some subset and $\mathcal{A}$ beats many nodes that beat a king of that subset, then $\mathcal{A}$ is an SE winner.

Lemma 1 (Kings Except for a T subset). *Let $\mathcal{A}$ be a node in a tournament G and let T be a subset of $N^{in}(\mathcal{A})$ of size $|T| = 2^k$ for some k. Suppose that $\mathcal{A}$ is a king in $G \setminus T$ and $|N^{out}(\mathcal{A})| \geq |N^{in}(\mathcal{A})|$. Let t be a king in T with outdegree in T at least $\lfloor |T|/2 \rfloor$. Suppose that $|N^{in}(t) \cap N^{out}(\mathcal{A})| \geq |T|$. Then $\mathcal{A}$ is an SE winner.*

The key observation in proving Lemma 1 is that t can win an SE tournament over just the subgraph consisting of T in $\log |T|$ rounds. At the same time, there are at least $|T|$ nodes in $N^{out}(\mathcal{A})$ that beat t. In the worst case, these cannot eliminate any other nodes in $N^{in}(\mathcal{A})$ so they must be matched against each other for $\log |T|$ rounds as well. However, given the size, we are guaranteed that at least 1 will survive to eliminate t. At this point, $\mathcal{A}$ will be a king of high outdegree over the induced subgraph. The technical details of the proof proceed by induction on the size of T. Lemma 1 is used in the proof of our main theorem below. We highlight its use in the intuitive sketch.

We now address the main question of this paper - what can we show when a matching from $V \setminus \mathcal{H}^{in}(\mathcal{A})$ to $\mathcal{H}^{in}(\mathcal{A})$ exists?

Theorem 2 (Not a King but Matching into $\mathcal{H}^{in}(\mathcal{A})$). *There exists a constant n_0 such that for all $n \geq n_0$ the following holds. Let $G = (V, E)$ be a tournament graph on n nodes, $\mathcal{A} \in V$. Suppose there is a matching M from $V \setminus \mathcal{H}^{in}(\mathcal{A})$ onto $\mathcal{H}^{in}(\mathcal{A})$ of size K. If $K \leq (n-6)/7$, then $\mathcal{A}$ is an SE winner.*

The key restriction in this Theorem concerns the number of higher ranked players who beat a player, not the actual rank of that player. However, we are able to apply the fact that a player of rank k has outdegree at least $(n-k-1)/2$ to obtain a nice corollary for large tournament graphs: Any one of the top 19% of the nodes are SE winners, provided there is a matching onto the nodes of higher outdegree.

Corollary 1. *There exists a constant n_0 so that for all tournaments G on $n > n_0$ nodes the following holds. Let $\mathcal{A}$ be among the top $(6n+7)/31 \geq .19n$ highest outdegree nodes. If there is a matching from $V \setminus \mathcal{H}^{in}(\mathcal{A})$ onto $\mathcal{H}^{in}(\mathcal{A})$, then $\mathcal{A}$ is an SE winner.*

3.1 Intuition

We now give an intuitive sketch about how one might go about proving Theorem 2. The overall strategy of our proof is to set up the first round of the SE tournament, so that all of the high outdegree nodes that beat $\mathcal{A}$ are eliminated, and in the remaining tournament, $\mathcal{A}$ is a king over almost the entire graph, so that Lemma 1 can be applied.

At first glance, one might try to build the first round by using the existing matching, M, from $V \setminus \mathcal{H}^{in}(\mathcal{A})$ to $\mathcal{H}^{in}(\mathcal{A})$ and then finding some maximal matching M' from $N^{out}(\mathcal{A}) \setminus M$ to $N^{in}(\mathcal{A}) \setminus M$. The matching M' will guarantee that as many elements as possible of $N^{out}(\mathcal{A})$ will survive to compete in the second round. To complete round 1, the potentially remaining nodes in $N^{out}(\mathcal{A}) \setminus (M \cup M')$ should be matched amongst themselves, and the same for $N^{in}(\mathcal{A}) \setminus (M \cup M')$ in a matching called M''.

$\mathcal{A}$ is initially a king in G over any node with no larger outdegree than it (i.e. $V \setminus \mathcal{H}^{in}(\mathcal{A})$). However, if we do not create the matching M'' above carefully $\mathcal{A}$ may no longer be a king over the sources of M. Even worse, some source of M may lose all of the nodes that can potentially beat, and might become a Condorcet winner in the graph induced by the winners of round 1. This is demonstrated in Figure 4. In this example, we would like to fix the bracket for P_2, the second strongest player. P_3 can beat P_1, but only P_{n-1} and P_n beat P_3. If we use any matching of $N^{out}(P_2)$ that does not match P_n with P_{n-1}, P_3 will be a Condorcet winner in round 2, and P_2 cannot win.

The failure of this example motivates our approach. We begin our construction of round 1 as before. We use the perfect matching M from $V \setminus \mathcal{H}^{in}(\mathcal{A})$ to $\mathcal{H}^{in}(\mathcal{A})$ and M', a maximal matching M' from $N^{out}(\mathcal{A}) \setminus M$ to $N^{in}(\mathcal{A}) \setminus M$. At this point, we want to guarantee that as many of the sources of M as possible are still covered by winners of round 1. We start by finding the set T of sources of M that are not currently beaten by some source in M', or by $\mathcal{A}$. Because these nodes are all of lower outdegree than $\mathcal{A}$, we can argue that there is some subset S which is a subset of $N^{out}(\mathcal{A}) \setminus (M \cup M')$ that covers T. We use a greedy approach (Algorithm 1) to match up the nodes of S in round 1 so that the winners of this matching cover as many nodes of T as possible. We are able to show (in Lemma 2) that the set U of nodes of T that are not covered by the first round winners from S is very small: it has size at most $O(\sqrt{|T|})$. This will allow us to show that we can eliminate U in later rounds.

We design the next rounds using Lemma 1. To do this, we use the largest outdegree node t in T and find a set P, the size of which is a power of 2, that contains both t and U. The final requirements of Lemma 1 are that $\mathcal{A}$ beats at least as many first round winners outside P as it

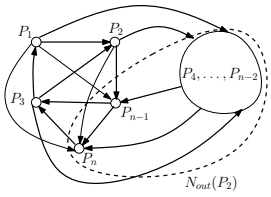

Figure 4: An example where an arbitrary matching of $N^{out}(P_2)$ is likely to fail.

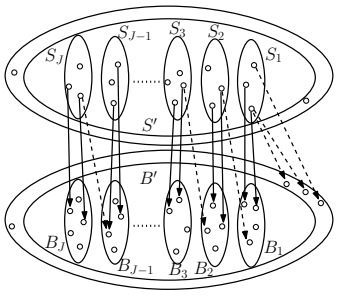

Figure 5: The construction of the sets S_i and B_i in Theorem 3.

loses to (which we show using Theorem 3) and that the number of nodes from $N^{out}(A)$ that beat t and survive round 1 is at least $|P|$. To fulfill this last requirement, we add an extra iteration (for $q = 1$) in Algorithm 1 which constructs the first round matching of S so that enough nodes that beat t survive round 1.

Summary. We create the first round of the tournament by using M, a maximal matching M' from the remaining nodes of $N^{out}(\mathcal{A})$ to the remaining nodes of $N^{in}(\mathcal{A})$, and a greedily selected perfect matching M'' on S. Many sources of M'' beat t, and almost all of T is covered by the sources of M''. This does not fully specify the first round matching. A few nodes may remain unmatched, specifically $N^{in}(\mathcal{A}) \setminus (M \cup M')$, $N^{out}(\mathcal{A}) \setminus (M \cup M' \cup M'')$ and $\mathcal{A}$ itself. The final details are included in the proof sketch at the end of this section. The goal of the first round matching is to ensure that the requirements of Lemma 1 are met and the remaining rounds of the tournament can be completed so that $\mathcal{A}$ wins.

3.2 Technical details.

With the above overview of the proof technique, we now introduce the necessary lemmas. As an SE tournament is a series of $\log n$ matchings, these lemmas are about the existence of matchings with desirable properties. The first is a very general result that can be specfically applied to lower-bound how large a matching can be found from $N^{out}(\mathcal{A})$ to $N^{in}(\mathcal{A}) \setminus \mathcal{H}^{in}(\mathcal{A})$.

Theorem 3 (Large Matching). *Let $h \in \mathbb{Z}$, possibly negative. Let S and B be disjoint sets such that $\forall X \subset B$, $|E(S, X)| \geq \binom{|X|}{2} - h|X|$. Then there exists a matching between S and B of size at least $\frac{|B|-2h-1}{2}$.*

Proof. Recall that M is a maximal matching from a set S to a set B if and only if there are no augmenting paths from the unmatched elements of S to the unmatched element of B. Our proof will proceed by using the large number of edges from S to any subset X of B to lower-bound the size of the matching.

Let M be a maximal matching from S to B. We refer to the sources of M as S' and the sinks as B'. We iteratively build up a family of sets S_j and B_j that consist of augmenting paths from the unmatched nodes in B.

Algorithm 1 Greedy Matching

```
Input: G = (V, E) a tournament and S, T ⊂ V, t ∈ V; Output: Matching M
Let A_1 = N_S^in(t), U_1 = T, i = 1, L_0 = ∅, M = ∅.
for q = 1, 2 do
    while |A_i| ≥ 2 do
        Let x_i, y_i ∈ A_i have larger outdegree to U_i than all the other elements in A_i; x_i beats y_i.
        M ← M ∪ {(x_i, y_i)}
        L_i ← L_{i-1} ∪ {y_i}
        U_{i+1} ← U_i \ N^out(x_i)
        A_{i+1} = ∪_{v∈U_{i+1}} N_{A_i}^in(v) \ L_i
        i ← i + 1
    end while
    A_i = ∪_{v∈U_i} N_S^in(U_i)
end for
```

Let S_1 be the subset of S' which contains all nodes with edges to $B \setminus B'$. Let B_1 be the nodes matched to S_1 by M. Now, we inductively define S_j as the nodes in $S' \setminus \cup_{i=1}^{j-1} S_i$ that have edges to B_{j-1}, where B_{j-1} are the nodes matched to S_{j-1} by M.

This process can be repeated up to some index $J + 1$ such that there are no more nodes in $S' \setminus \cup_{i=1}^{J} S_i$ with edges to B_J. Let $\bar{S} = \cup_{i \leq J} S_i$ and $\bar{B} = (B \setminus B') \cup (\cup_{i \leq J} B_i)$.

First, note that there are no edges from $S \setminus S'$ to $\bar{B}$ since M is maximal. If there were, we would have an augmenting path. Therefore, all edges into $\bar{B}$ come from $\bar{S}$. The number of edges from $\bar{S}$ into $\bar{B}$ is at most $|\bar{B}||\bar{S}|$ (the number of edges in a complete bipartite graph) and at least $|\bar{B}|(|\bar{B}| - 1 - 2h)/2$ by the Theorem statement. Thus, we can conclude that

$$|M| = |B \setminus \bar{B}| + |\bar{S}| \geq$$

$$(|B| - |\bar{B}|) + \frac{(|\bar{B}| - 1 - 2h)}{2} \geq \frac{(|B| - 1 - 2h)}{2}.$$

□

Theorem 3 is used in the proof of Theorem 2 to argue about a lower bound on the size of $N^{out}(\mathcal{A})$ after the first round. An example application of this theorem is to set S to $N^{out}(\mathcal{A})$ and B to $N^{in}(\mathcal{A}) \setminus (M \cup \mathcal{H}^{in}(\mathcal{A}))$. Here, the conditions of the Theorem are met: we can show that for every subset X, $E(S, X) \geq \binom{|X|}{2} + |X|$ because every vertex in B beats $\mathcal{A}$ and is of lower outdegree than $\mathcal{A}$.

The other very important part of our proof is Algorithm 1. As mentioned earlier, it is a greedy way of creating a matching in a set S such that the sources cover many elements in a set T. It iteratively finds the source in S that covers the most uncovered elements of T and matches it with a vertex that it beats. The first iteration of the loop deals with an element t that is a king over T. This loop only considers the subset of S that beats t and guarantees that at least half of the nodes that beat t in S

are preserved as sources. At any time in the algorithm, U_i is the set of the nodes that are currently not covered by the sources of the matching M, A_i is the set of sources that beat any element in U_i, and L_i is the set of nodes that lose in M and are excluded from A_i.

We want to lower-bound the size of the generated cover. The main idea of the proof is that we initially have many edges from S to T, and specifically at least $\binom{|X|+1}{2}$ to each $X \subseteq T$. If we consider the first pair (x_1, y_1) added to M, then we can say x_1 covers k elements of T. Therefore, we now need to cover only a subset of size $|T| - k$ which has at least $\binom{|T|-k}{2}$ edges into it. However, this may include edges from y_1. When we remove y_1, we may lose up to $|T|$ edges. The key observation is that for the pair (x_2, y_2), y_2's outdegree is upper-bounded by x_1 so we are able to bound the number of edges lost by the matching as the number of vertices currently covered plus $|T|$. We then show that there will always be enough edges and sources to increase the size of the matching until at most $2\sqrt{|T|} + 1$ nodes remain uncovered.

Lemma 2. *Let $G = (V, E)$ be a tournament graph. Let $S \subseteq V$ and $T \subseteq V$ be disjoint sets such that for all $X \subseteq T$, the number of edges from S to X is at least $\binom{|X|+1}{2}$. Let $t \in V$ be given. Algorithm 1 generates a matching, M, in S such that at least $|T| - 1 - 2\sqrt{|T|}$ nodes in T are beaten by at least one source in M and at least $(in_S(t) - 2)/2$ of the sources also beat t.*

Proof. We need to define some additional concepts for the proof. The first is the set of covered nodes at iteration i, C_i, where $C_1 = \emptyset$. C_i is exactly $T \setminus U_i$ (so $|T| = |C_i| + |U_i|$). Let $d_i = |C_{i+1}| - |C_i|$ be the number of new nodes covered by iteration i. Our goal is to lower-bound the size of $|C_i|$ when the algorithm quits.

Consider the first execution of the WHILE loop. Let i_0 be the iteration at which the loop exits. This loop greedily covered T but only used vertices that also beat t. We will lower-bound the number of edges that remain from all unmatched sources in S (the set A_{i_0}) to U_{i_0}. At this point, $|C_{i_0}| = \sum_{j=1}^{i_0} d_j$. The number of edges from L_{i_0} to U_{i_0} is at most $|T| - |C_{i_0}| + \sum_{j=1}^{i_0-1} d_j \leq |T|$ since we picked the nodes so that $out_{U_i}(y_i) \leq out_{U_{i-1}}(x_{i-1}) = d_{i-1}$, and $out_{U_{i_0}}(y_1) \leq |U_{i_0}|$. Thus we can obtain a lower bound on the number of edges between A_{i_0} and U_{i_0}: $|E(A_{i_0}, U_{i_0})| \geq \binom{|U_{i_0}|+1}{2} - |T|$.

Let $j > i_0$ be any round in the second WHILE loop. As above, $|C_j| = |C_{i_0}| + \sum_{k=i_0+1}^{j} d_k$ and the number of edges from L_j to U_j is at most

$$|U_j| + |T| + \sum_{k=i_0+1}^{j-1} d_k = 2|T| - |C_j| + |C_j| - |C_{i_0}| \leq 2|T|.$$

We can lower-bound the number of usable edges from A_j to U_j as

$$|E(A_j, U_j)| \geq \binom{|U_j|+1}{2} - 2|T| \geq$$

$$(|T|^2 + |C_j|^2 - (2|T| + 1)|C_j| - 3|T|)/2.$$

The second WHILE loop exits when $|A_j| \leq 1$. Therefore, when the algorithm finishes, $|A_j| \leq 1$ and $|E(A_j, U_j)| \leq |U_j| = |T| - |C_j|$. We have:

$$(|T|^2 + |C_j|^2 - (2|T| + 1)|C_j| - 3|T|)/2 \leq |T| - |C_j|,$$

This can be simplified as follows.

$$|C_j|^2 - (2|T| - 1)|C_j| - 5|T| + |T|^2 \leq 0.$$

$$|C_j| \geq |T| - 1/2 - \sqrt{|T|^2 - |T| + 1/4 + 5|T| - |T|^2} =$$

$$|T| - 1/2 - \sqrt{4|T| + 1/4} \geq |T| - 1 - 2\sqrt{|T|}.$$

That is, the number of covered nodes is at least $|T| - 1 - 2\sqrt{|T|}$. After round i_0 we have at least i_0 sources in M covering t and at least $in_S(t) - 2i_0 - 1$ nodes of $N_S^{in}(t)$ that were not used in creating the rest of the matching because they did not cover any element of U_{i_0}. Match these among themselves to obtain at least $i_0 + \lfloor (in_S(t) - 1 - 2i_0)/2 \rfloor \geq (in_S(t) - 2)/2$ sources of the matching that are inneighbors of t. Complete the matching M from S to S by matching the rest of the nodes of S arbitrarily. □

The bounds on the greedy matching algorithm are only positive if $|T| > 5$. We don't want our bounds in Theorem 2 to depend on the size of the matching into $\mathcal{H}^{in}(\mathcal{A})$. In the full version, we fix this problem with a technical lemma that allows one to artifically boost the size of T to guarantee that the above process will always work.

Now we give a more technical proof of Theorem 2. Although the proof has most of the key details, it does not contain all of them. In particular, it assumes that that the indegree of node $\mathcal{A}$ coming from the sources of M is large enough. This assumption is lifted in the full version of the paper.

Proof sketch of Theorem 2: This proof proceeds by constructing the first round matching in stages. First, we will use M, the matching given by the theorem statement, and construct M', a maximal matching. Next, we show how to match $\mathcal{A}$ and construct the covering of the sources of M using Algorithm 1. Finally, we argue that the constructed first round matching satisfies the requirements of Lemma 1.

For simplicity, let $A = N^{out}(\mathcal{A})$ and $B = N^{in}(\mathcal{A})$. We divide the sources of M onto $\mathcal{H}^{in}(\mathcal{A})$ into two sets, A_T and B_T, where A_T are the sources of M in A while B_T are the sources in B. We can also divide $\mathcal{H}^{in}(\mathcal{A})$ into two sets, H_1 and H_2, where H_1 are the nodes matched to A_T and H_2 are matched to B_T by M. In order to later argue about the size of matchings, let $|A_T| = |H_1| = h$ and $|B_T| = |H_2| = k$. This means that K, the size of M is exactly $k + h$.

Let $B_{\text{rest}} = B \setminus (B_T \cup \mathcal{H}^{in}(\mathcal{A}))$ be the nodes who beat $\mathcal{A}$ and are not part of M. Take M' to be any maximal matching from $A \setminus A_T$ to B_{rest}. We want to argue about the size of M' by using Theorem 3. First, note that $|B_{\text{rest}}| = |B| - k - K$. Now, since we removed A_T, of

size h, we can only say that every node b in B_{rest} has at least $out_B(b)+1-h$ inneighbors from $A \setminus A_T$. Therefore, by Theorem 3, $|M'| \geq (|B| - K - k - 2h + 2 - 1)/2 = (|B| - 2K - h + 1)/2$. We will use this fact later when arguing about the outdegree of $\mathcal{A}$ after the first round.

Finally, note that B_{rest} consists only of lower ranked nodes than $\mathcal{A}$, so every node in B_{rest} has some source of M' or A_T as an inneighbor.

(Matching $\mathcal{A}$ to some node.) Consider the currently unmatched portion of A. Call this $A_{\text{rest}} = A \setminus (A_T \cup M')$. If there is some $a' \in A_{\text{rest}}$, then match $\mathcal{A}$ to a'. If A_{rest} is empty, then we can argue that $|M'| > 1$ since $|A \setminus A_T| = |A| - h \geq (n-K)/3 - h \geq (n - 4K)/3 > 1$. Since $M' > 1$, we can dislodge any edge (a', b') from M' and match $\mathcal{A}$ to a'. After removing a', the lower bound for $|M'|$ goes down by 1: $|M'| \geq (|B| - 2K - h - 1)/2$.

(Creating a matching of $A_{\text{rest}} \setminus \{a'\}$.) We now use Algorithm 1 to cover B_T. Let $S = A_{\text{rest}} \setminus \{a'\}$ and T be the subset of B_T consisting of the nodes that do not have inneighbors among the sources of M' and A_T. For simplicity in this proof we assume that $|T|$ and hence $|B_T|$ is large enough.

Every subset X of the nodes of T has at least $\binom{|X|}{2} + 2|X| - |X| = \binom{|X|}{2} + |X|$ inneighbors in S since each node in X can have lost at most one inneighbor, a'. Let $t \in B_T$ be the node with highest outdegree in B_T. Run Algorithm 1 on S, T, t. This outputs a matching M'' on the nodes of S that covers all of T except for a subset, U, of size at most $1 + 2\sqrt{|T|}$. There are also at least $in_S(t)/2 - 1$ sources of M'' that beat t. This completes the first round matching. Let G' be the graph induced by the surviving nodes.

(Handling U.) We will construct P, a subset of T, such that P contains U, and t is a king over P who beats at least half of P. We selected t so that it is a king in T. Therefore, there is a subset of at most $|U|$ nodes in its outneighborhood in T that cover U. We can add these nodes together with enough other nodes of $N_T^{out}(t)$ to P so that $|P|$ is a power of 2 and t is a king in P that beats at least half of P. This is possible since U is very small compared to T.

We can assume that the size of P is 2^c where 2^c is the closest power of 2 greater than $3 + 4\sqrt{|T|}$, as we may need as many as $|U| \leq 1 + 2\sqrt{|T|}$ extra nodes added to P to guarantee that t is a king over P. We can further conclude that $|P| \leq 5 + 8\sqrt{|T|}$ since we can at most double $3 + 4\sqrt{|T|}$ to make $|P|$ be a power of 2.

From Algorithm 1 we know that at least $in_S(t)/2 - 1 \geq (|B_T| - 1)/4 - 1$ inneighbors of t from S are in G'. Since we assumed that B_T is large enough, we have $(|B_T| - 1)/4 - 1 \geq 5 + 8\sqrt{|T|}$. Hence there exists a subset of the surviving nodes of $N_S^{in}(t)$ of size at least $|P|$. The requirements of Lemma 1 are satisfied if $out_{G'}(\mathcal{A}) \geq in_{G'}(\mathcal{A})$. We prove this below and thus show that $\mathcal{A}$ is an SE winner.

(Showing that $out_{G'}(\mathcal{A}) \geq in_{G'}(\mathcal{A})$.) The number of nodes of $N^{out}(\mathcal{A})$ that survive the first round is at least $\lfloor (|A| + |M'| + |A_T| - 1)/2 \rfloor$. The number of nodes of $N^{in}(\mathcal{A})$ that survive is at most $\lceil (|B| - |A_T| - |M'|)/2 \rceil$. It suffices to show that $|A| + |M'| + |A_T| - 1 \geq |B| - |A_T| - |M'|$. Recall that $|M'| \geq (|B| - 2K - h - 1)/2$ so we must only show that $|A| + |B| - 2K - h + 2h - 2 \geq |B|$, or that $|A| - 2K + h - 2 \geq 0$. Since $|A| \geq (n-K)/3$ it suffices to show that $(n - K) \geq 6K + 6$, or that $K \leq (n-6)/7$, which is true by assumption. □

4 Conclusions

In this paper, we have shown that the existence of back-matchings can allow for an SE tournament to be manipulated in favor of any of the top 19% of players. The Braverman-Mossel model for tournament generation shows that back-matchings exist even when the noise in each match is very low ($O(\frac{\log n}{n})$). The question of the computational difficulty of manipulating SE tournaments in favor of a specific player remains open, but our result shows that many common examples can be efficiently manipulated in polynomial time using any algorithm for maximum matching. The fastest algorithms for matching run in $\tilde{O}(m\sqrt{n})$ time ([Micali and Vazirani, 1980] reimplementing [Edmonds, 1965]) and in $\tilde{O}(n^{2.376})$ time [Mucha and Sankowski, 2004]. Possible future work includes finding even more general conditions under which a winning bracket can be found for a player, as well as trying to reduce the dependence our method has on the number of players.

Acknowledgments

The authors are grateful for the detailed comments from the anonymous reviewers. The first author was supported by the NDSEG and NSF Fellowships. The second author was supported by the National Science Foundation under Grant #0937060 to the Computing Research Association for the CIFellows Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Computing Research Association.

References

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

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

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

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

[Chen and Hwang, 1988] Robert Chen and F. K. Hwang. Stronger players win more balanced

knockout tournaments. *Graphs and Combinatorics*, 4(1):95–99, 1988.

[Chevaleyre *et al.*, 2007] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. *SOFSEM*, 4362:51–69, 2007.

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

[Edmonds, 1965] J. Edmonds. Paths, trees, and flowers. *Canad. J. Math.*, 17:449–467, 1965.

[Erdős and Rényi, 1964] P. Erdős and A. Rényi. On random matrices. *Publ. Math. Inst. Hung. Ac. Sci.*, 8:455–561, 1964.

[Fischer *et al.*, 2009] F. Fischer, A.D. Procaccia, and A. Samorodnitsky. A new perspective on implementation by voting trees. In *Proc. EC*, pages 31–40, 2009.

[Hazon *et al.*, 2008] N. Hazon, P.E. Dunne, S. Kraus, and M. Wooldridge. How to rig elections and competitions. In *Proc. COMSOC*, 2008.

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

[Hwang, 1982] F. K. Hwang. New concepts in seeding knockout tournaments. *The American Mathematical Monthly*, 89(4):235–239, 1982.

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

[Micali and Vazirani, 1980] S. Micali and V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In *Proc. FOCS*, pages 17–27, 1980.

[Mucha and Sankowski, 2004] M. Mucha and P. Sankowski. Maximum matchings via gaussian elimination. In *FOCS*, pages 248–255, 2004.

[Russell and Walsh, 2009] T. Russell and T. Walsh. Manipulating tournaments in cup and round robin competitions. In *Proc. ADT*, pages 26–37, 2009.

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

[Stanton and Vassilevska Williams, 2011] I. Stanton and V. Vassilevska Williams. Rigging tournament brackets for weaker players. *UC Berkeley EECS Tech. Report*, May 2011.

[Vassilevska Williams, 2010] V. Vassilevska Williams. Fixing a tournament. In *Proc. AAAI*, pages 895–900, 2010.

[Vu and Shoham, 2010a] T. Vu and Y. Shoham. Fair seedings in knockout tournaments. *ACM Transactions on Intelligent Systems and Technology*, 2010.

[Vu and Shoham, 2010b] T. Vu and Y. Shoham. Optimal seeding in knockout tournaments. In *Proc. AAMAS*, pages 1579–1580, 2010.

[Vu *et al.*, 2010] T. Vu, N. Nazon, A. Altman, S. Kraus, Y. Shoham, and M. Wooldridge. On the complexity of schedule control problems for knock-out tournaments. *JAIR*, 2010.