How to Make Knockout Tournaments More Popular?

Juhi Chaudhary, Hendrik Molter, Meirav Zehavi

Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel
juhic@post.bgu.ac.il, molterh@post.bgu.ac.il, meiravze@bgu.ac.il

Abstract

Given a mapping from a set of players to the leaves of a complete binary tree (called a *seeding*), a *knockout tournament* is conducted as follows: every round, every two players with a common parent compete against each other, and the winner is promoted to the common parent; then, the leaves are deleted. When only one player remains, it is declared the winner. This is a popular competition format in sports, elections, and decision-making. Over the past decade, it has been studied intensively from both theoretical and practical points of view. Most frequently, the objective is to seed the tournament in a way that "assists" (or even guarantees) some particular player to win the competition. We introduce a new objective, which is very sensible from the perspective of the directors of the competition: maximize the profit or popularity of the tournament. Specifically, we associate a "score" with every possible match, and aim to seed the tournament to maximize the sum of the scores of the matches that take place. We focus on the case where we assume a total order on the players' strengths, and provide a wide spectrum of results on the computational complexity of the problem.

1 Introduction

A *knockout (or single-elimination) tournament* is the most popular competition format in sports (Horen and Riezman 1985; Connolly and Rendleman 2011; Groh et al. 2012). Here, roughly speaking, the players are paired up to compete against each other in rounds, where, at each round, the losers are knocked out, until only one player (the winner) remains. For an illustrative example, consider any major tennis tournament or the knockout stage of the football world cup. Notably, knockout tournaments are common in other fields as well, such as elections and decision-making (Suksompong 2021; Tullock 1980; Rosen 1986; Laslier 1997).

More formally, we are given n *players* (where, for simplicity, n is a power of 2) and a *seeding* that determines how to label the n leaves of a complete binary tree with the players. Given a seeding, the competition is conducted in rounds: As long as the tree has at least two leaves, every two players with a common parent in the tree play against each other, and the winner is promoted to the common parent; then, the leaves of the tree are deleted from it. Eventually, only one player remains, and this player is declared the winner.

Over the past decade, knockout tournaments have been studied intensively from both theoretical and practical points of view. Most of the attention has been given to the objective of making a specific player win, particularly by selecting a seeding that is considered advantageous to this player, as well as bribing some of the players (see, e.g., the surveys by Suksompong (2021) and Williams and Moulin (2016)). Here, to decide whether a seeding is advantageous, we assume to possess *predictions* (deterministic or probabilistic) on the winners of (all or some of) the possible games or matches (we use these two terms interchangeably). In particular, when the aforementioned information is deterministic and complete (i.e., for every possible match involving two players, we "know" who will be the winner) and our only influence on the competition is by picking the seeding, the problem is known as the TOURNAMENT FIXING problem. This problem and its various variations have been extensively studied from the viewpoints of classical complexity, parameterized complexity, and structural analysis (see, e.g., (Gupta et al. 2019, 2018a,b; Kim and Williams 2015; Aziz et al. 2018; Vu, Altman, and Shoham 2009; Williams 2010; Zehavi 2023; Ramanujan and Szeider 2017; Stanton and Williams 2011b,a; Kim, Suksompong, and Williams 2017; Manurangsi and Suksompong 2023); this list is illustrative rather than comprehensive).

One aspect of our contribution is conceptual: we introduce a new objective, which is, perhaps, more sensible from the perspective of the directors of the competition: maximize the profit or popularity of the tournament. Indeed, this is the main reason behind commercials and promotions for sports competitions and a subject of active discussion in the media. More formally, the input consists of n players and, for every possible match between two players, the winner of that match. Additionally, for every possible match between two players and the round in which it is to be conducted, we have the *value* of that match in that round, represented by an integer. Here, the value can correspond to the profit, popularity, or any other measure (or combination thereof) associated with the match in that round. We expect matches conducted in later rounds to be more profitable/popular; yet, when rounds do not affect this matter, the value function is said to be *round-oblivious*. The latter is plausible in many cases, such as local derbies or games between two rivaling teams or players, which roughly maintain their excitement

regardless of the tournament round in which it occurs (Hall, O'Mahony, and Vieceli 2010; Chmait et al. 2020). The task is to compute a seeding that maximizes the sum of the values of the matches that take place in all rounds. Clearly, this problem can be naturally generalized to scenarios where the predictions are probabilistic or/and partial, to maximize the *expected* value.

Being the first study on the computational complexity of this problem, we focus on the simplest and most basic case, where the predictions correspond to a linear ordering of the players, say, from weakest to strongest. This is also the most realistic case in terms of existing inputs. Indeed, a linear ordering can be simply obtained by taking an existing ranking of the players or calculating a number based on their previous performances (e.g., as is done in tennis (ESPN 2023b,a)). However, in many cases, it is unclear how to obtain more complicated, non-linear assessments of outcomes. Moreover, intuitively, the deterministic model can serve as a "reasonable approximation" for settings where linear player rankings are available and the winning probabilities are either unknown or assumed to be close to 1/0. Henceforth, we refer to our problem—with a linear ordering of the players—as TOURNAMENT VALUE MAXIMIZATION; a formal definition can be found in Section 2. We note that for the TOURNAMENT FIXING problem, the case of a linear ordering makes the problem trivial, since then, for any seeding, the same player (being the strongest one) wins.

Other Related Works. There is a huge body of research on TOURNAMENT FIXING and related tournament manipulation problems; we gave an illustrative list in the introduction. However, to the best of our knowledge, there is no prior work on TOURNAMENT VALUE MAXIMIZATION, with the following exception. Dagaev and Suzdaltsev (2018) investigated a highly restricted case of our setting, where every player has a distinct strength value, and the value of a game is determined by (a linear combination) of its "quality" (the sum of strengths of the involved players) and its "intensity" (the absolute difference of the strengths of the involved players). They characterize cases where either a "close" seeding is optimal, a "distant" seeding is optimal, or every seeding is optimal. In particular, this implies that their restricted cases are trivially solvable in linear time.

Our Contributions. We introduce TOURNAMENT VALUE MAXIMIZATION and analyze its computational complexity. Due to space constraints, the proofs of some results (marked with $\star$) are deferred to the full version of this paper (Chaudhary, Molter, and Zehavi 2023). In Section 3, we prove that it is NP-hard (as well as APX-hard) in two highly restricted scenarios: when all game values are 0 or 1, or when the game-value function is round-oblivious and there are 3 distinct game values. Here, the proofs are based on non-trivial reductions from MAX $(2,3)$-SAT.

Nevertheless, we provide a wide spectrum of positive results in Section 4. First, we provide a simple $(1/\log n)$-factor approximation algorithm based on the computation of a maximum-weight matching in a graph. Second, we identify a large family of game-value functions that give rise to efficient algorithms: a quasipolynomial-time algorithm (i.e., with runtime $n^{\mathcal{O}(\log n)}$). Intuitively, this family consists of game-value functions that allow the total value of a tournament to be computed from the information on the number of wins of each player. Here, our main tool is the introduction of the concept of *open* and *closed subtournaments* (given a partial seeding). We use it to design a dynamic programming algorithm. Moreover, we identify a natural restriction that facilitates a simple greedy algorithm. Here, each player is assigned a popularity value, and the value of a game is equal to the popularity value of the winning player. Additionally, we assume that there are only two different player popularity values. Intuitively speaking, players can be categorized as either "popular" or "unpopular".

Still regarding positive results, now at the parameterized complexity front, our contribution is twofold. First, we consider the setting where each player is once again assigned a popularity value, and the value of a game is equal to the popularity value of the winning player. Note that the popularity values naturally define an ordering of the players, from most to least popular. For this setting, we consider a natural distance measure between the strength ordering and the ordering given by the popularity values of the players (for a formal definition, see Section 4). For this setting, we present a *fixed-parameter algorithm* (i.e., an algorithm with a runtime of the form $f(k) \cdot n^{\mathcal{O}(1)}$ for a function f depending only on k). The motivation for considering this parameter

Restriction	Hardness Results	Algorithmic Results
unrestricted	NP-hard and APX-hard for 2 game values (Theorem 5)	-
round-oblivious	NP-hard and APX-hard for 3 game values (Theorem 6)	$(1/\log n)$-factor approximation (Theorem 7)
		FPT w.r.t. the size of a minimum influential set of players (Theorem 15)
win-count oriented	-	$n^{\mathcal{O}(\log n)}$-time algorithm (Theorem 8)
player popularity-based (implies round oblivious and win-count oriented)	-	linear-time algorithm for 2 player values (Theorem 10)
		FPT w.r.t. the disagreement between the player popularity values and the strength ordering (Theorem 12)

Table 1: Table of Results.

stems from the common observation that the popularity and fascination surrounding players are often highly correlated to their capabilities. Second, for (general) round-oblivious value functions, we consider the parameter k as the minimum size of a so-called *influential set of players*. Roughly speaking, we define an influential set as a set of players such that every match that does not involve any of them has value 0 (for a formal definition, see Section 4). We design a fixed-parameter algorithm for this parameter as well. To this end, we adapt the recent algorithm of Zehavi (2023) for TOURNAMENT FIXING parameterized by the feedback vertex set number of the *prediction graph* of the input, being the complete digraph obtained by having an arc from a player a to a player b if a is predicted to beat b. The motivation to consider this parameter is that, quite often, only a small set of players are truly profitable or popular.

Our main results are summarized in Table 1. We hope that our work will open up the door for further studies of the maximization of the profit or popularity of tournaments, and present some directions for further research in Section 5.

2 Problem Setting and Preliminaries

In our tournament setting, we are given a set N of $n = |N|$ *players* (where, for simplicity, we assume that n is a power of 2). Furthermore, we assume to possess deterministic *predictions* on the winners of all possible matches. Generally, this is modeled by a so-called *tournament graph*, a directed graph that has the set of players as vertices and where we have one arc between each pair of players. As mentioned in the introduction, our work focuses on the special case where we have a linear ordering on the players that defines their relative strength. We assume that the ordering is from strongest to weakest player. In other words, we consider the setting where the tournament graph is acyclic. More formally, we define *players* as natural numbers, and we say that a player i *beats* a players j if $i > j$.

A *seeding* that determines how to label the n leaves of a complete binary tree with the players. Given a seeding, the competition is conducted in rounds as follows. As long as the tree has at least two leaves, every two players with a common parent in the tree play against each other, and the winner is promoted to the common parent; then, the leaves of the tree are deleted from it. Eventually, only one player remains, and this player is declared the winner.

Formally, given a set $N \subseteq \mathbb{N}$ of players with $|N| = n = 2^{n'}$ for some $n' \in \mathbb{N}$, we define a *tournament seeding* for N as an injective function $\sigma : N \to [n]$, where for $k \in \mathbb{N}$ we denote $[k] = \{1, 2, \ldots, k\}$. Informally speaking, $\sigma(i)$ is the seed position of player i and corresponds to the ordinal position of the leaf player i is assigned to in a DFS-ordering of the leaves of a complete binary tree with n leaves.

For some $N' \subseteq N$ with $|N'| = 2^{n''}$ and a seeding σ for N, we denote the function $\sigma' : N' \to [|N'|]$ as a *partial seeding* for N', if for all players $i, j \in N'$, we have that $\sigma'(i) - \sigma'(j) = \sigma(i) - \sigma(j)$. We use a *game-value function* $v : N \times N \times \mathbb{N} \to \mathbb{Z}$ to quantify the value of a game. If player i plays against player j in round r, the value of this game is $v(i, j, r)$. The value of the tournament is the sum of the values of the games played. Formally:

Definition 1 (Tournament Value)**.** *Given a set of players N, a game-value function $v : N \times N \times \mathbb{N} \to \mathbb{Z}$, and a tournament seeding σ, the* tournament value V_σ *is defined as follows.*

- *If $N = \{i, j\}$ with $i > j$, then for every tournament seeding σ_0 for players i, j the tournament value V_{σ_0} is $v(i, j, 1)$ and player i wins the tournament.*
- *Assume $|N| = 2^n$ for some $n > 1$. Let $N_1 \subset N$ with $|N_1| = 2^{n-1}$ and let $N_2 = N \setminus N_1$ such that for all $i \in N_1$ and $j \in N_2$ we have that $\sigma(i) < \sigma(j)$. Let σ_1 be the partial seeding for the players in N_1 and let σ_2 be the partial seeding for the players in N_2. Let V_{σ_1} be the value of the subtournament of players in N_1 and let i_1 be its winner. Analogously, let V_{σ_2} be the value of the subtournament of players in N_2 and let i_2 be its winner. Then, the value of the tournament is $V_\sigma = V_{\sigma_1} + V_{\sigma_2} + v(i_1, i_2, n)$ and if $i_1 > i_2$, then i_1 wins the tournament, otherwise i_2 wins the tournament.*

The main problem we investigate asks whether we can find a seeding that guarantees a certain minimal value of the tournament. Our formal problem definition is the following, where we assume that the set of players in the input is sorted by the strength of the players (from strongest to weakest).

TOURNAMENT VALUE MAXIMIZATION	
Input:	A set $N \subset \mathbb{N}$ of players with $\|N\| = n = 2^{n'}$ for some $n' \in \mathbb{N}$, a game-value function $v : N \times N \times \mathbb{N} \to \mathbb{Z}$, and a target value V.
Question:	Is there a tournament seeding σ for the players in N such that the tournament value V_σ is at least V?

We will consider some natural restrictions on the game-value function. We call a game value v function *home team oblivious* if for all pairs of players i, j and all rounds r we have that $v(i, j, r) = v(j, i, r)$. We call a game value v function *round-oblivious* if for all pairs of players i, j and all rounds r, r' we have that $v(i, j, r) = v(i, j, r')$.

Furthermore, we make the following observations.

Observation 1 ($\star$)**.** *Let I be an instance of* TOURNAMENT VALUE MAXIMIZATION. *Let I' be the instance obtained from I by replacing the game-value function v with v', where for all $i, j \in N$ and for all $r \in \mathbb{N}$ we have that $v'(i, j, r) = \max\{v(i, j, r), v(j, i, r)\}$. It holds that I is a yes-instance if and only if I' is a yes-instance.*

Observation 1 allows us w.l.o.g. to only consider TOURNAMENT VALUE MAXIMIZATION instances with home team oblivious game-value functions. Furthermore, we can observe that we can add some constant to each game value without significantly changing the tournament.

Observation 2 ($\star$)**.** *Let I be an instance of* TOURNAMENT VALUE MAXIMIZATION. *Let I' be the instance obtained from I by replacing the game-value function v with v', where for all $i, j \in N$ and for all $r \in \mathbb{N}$ we have that $v'(i, j, r) = v(i, j, r) + c$ for some $c \in \mathbb{Z}$ and by replacing the target value V with $V' = V + (n - 1) \cdot c$. It holds that I is a yes-instance if and only if I' is a yes-instance.*

To find tractable cases, we investigate a restricted class of game-value functions that we call *win-count oriented*. Intuitively speaking, such a game-value function allows one to evaluate the value of the tournament by looking at each player and the number of games they won.

Definition 2 (Win-Count-Oriented Game-Value Function)**.** *A game-value function $v : N \times N \times \mathbb{N} \to \mathbb{Z}$ is* win-count-oriented *if there is a* player evaluation function $p : N \times \mathbb{N} \to \mathbb{Z}$ *such that for every seeding σ the tournament value V_σ equals $\sum_{i \in N} p(i, w_\sigma(i))$, where $w_\sigma(i)$ denotes the number of wins of player $i \in N$ when tournament seeding σ is used.*

We give an alternative characterization of win-count-oriented game-value functions. We show that the game-value function is win-count oriented if and only if the function value only depends on the round and the winning player.

Proposition 3 ($\star$)**.** *A game-value function $v : N \times N \times \mathbb{N} \to \mathbb{Z}$ is win-count oriented if and only if there exists a function $v' : N \times \mathbb{N} \to \mathbb{Z}$ such that for all $i, j, r \in N \times N \times \mathbb{N}$ we have $v(i, j, r) = v'(\max(i, j), r)$.*

Another natural restriction is the setting where every player has a popularity value, and the value of a game equals the popularity value of the winning player. We call game-value functions of this setting *player popularity-based*.

Definition 3 (Player Popularity-Based Game-Value Function)**.** *A game-value function $v : N \times N \times \mathbb{N} \to \mathbb{N}$ is* player popularity-based *if there are* player popularity values v_i *for all players $i \in N$ such that for all $i, j \in N$ and $r \in \mathbb{N}$, we have $v(i, j, r) = v_{\max(i,j)}$.*

We can observe that player popularity-based game-value functions are exactly the win-count-oriented game-value functions that are also round oblivious.

Observation 4. *A game-value function is player popularity-based if and only if it is win-count oriented and round-oblivious.*

3 Hardness Results

In this section, we consider computational hardness and inapproximability for TOURNAMENT VALUE MAXIMIZATION and its optimization version. To show the APX-hardness, we give an L-reduction from MAX (2,3)-SAT, which is known to be NP-hard and APX-hard (Ausiello et al. 2012). In MAX (2,3)-SAT, we are given a Boolean formula ϕ such that each clause has exactly two literals and each variable appears in at most three clauses, and we are asked to find an assignment to the variables that satisfies the maximum number of clauses of ϕ.

Theorem 5 ($\star$)**.** TOURNAMENT VALUE MAXIMIZATION *is NP-hard and the optimization version of* TOURNAMENT VALUE MAXIMIZATION *is APX-hard for game-value functions that map to $\{0, 1\}$.*

Theorem 6 ($\star$)**.** TOURNAMENT VALUE MAXIMIZATION *is NP-hard and the optimization version of* TOURNAMENT VALUE MAXIMIZATION *is APX-hard for round-oblivious game-value functions that map to three distinct values.*

4 Algorithmic Results

A Polynomial-Time $(1/\log n)$-Approximation Algorithm. Here, we show the existence of a $(1/\log n)$-approximation algorithm for the optimization version of TOURNAMENT VALUE MAXIMIZATION with a round-oblivious game-value function that runs in polynomial time.[1] Formally, we prove the following theorem.

Theorem 7. *The optimization version of* TOURNAMENT VALUE MAXIMIZATION *with a round-oblivious game-value function admits a polynomial-time $(1/\log n)$-approximation algorithm.*

Proof. Consider Algorithm $\mathcal{A}$ that, given an instance T of the optimization version of TOURNAMENT VALUE MAXIMIZATION with the player set N and with a round-oblivious game-value function v, performs the following steps (1-5):

1. First, construct a complete undirected graph G that has N as its vertex set and an edge-weight function $w : E(G) \to \mathbb{N}$ defined as follows:
$$w(\{i, j\}) = \max(v(i, j), v(j, i)).$$
2. Compute a maximum weight matching $M_{\max} \subseteq E(G)$ in G.
3. Let $e_\ell = \{i, j\} \in M_{\max}$ denote the ℓth edge in $M_{\max}$ and let w.l.o.g. $v(i, j) \geq v(j, i)$. Then, define a seeding σ such that player i is seeded into position $2\ell - 1$ and player j is seeded into position 2ℓ.
4. Let $\overline{V}_{\max}$ denote the set of vertices that are not saturated by $M_{\max}$. For every $i \in \overline{V}_{\max}$, seed player i arbitrarily into any of the remaining seed positions of σ(which are, $\{2|M_{\max}| + 1, \ldots, n\}$).
5. Return σ.

Now, we claim that if $W_{\max}$ denotes the tournament value of T corresponding to the seeding σ (returned by Algorithm $\mathcal{A}$), then $W_{\max}$ is an $(1/\log n)$-approximation of the maximum achievable tournament value of T. Before we prove the approximation bound, note that Algorithm $\mathcal{A}$ runs in polynomial time, as maximum weight matchings can be computed in polynomial time (Galil 1986).

In the remainder, we show that Algorithm $\mathcal{A}$ is a $(1/\log n)$-approximation algorithm for TOURNAMENT VALUE MAXIMIZATION. Let $V^\star$ denote the highest achievable tournament value of T. Observe that we have
$$V^\star \geq W_{\max}.$$

Now, let $V^\star_r$ denote the value of games played in round $r \in [\log n]$ in the seeding that achieves tournament value $V^\star$. We have that
$$V^\star_r \leq W_{\max},$$
since otherwise the games played in round r would correspond to edges in G that form a matching with a weight larger than $W_{\max}$ (because the value of $W_{\max}$ is exactly equal to the weight of $M_{\max}$).

[1] Throughout this document, log refers to the base-2 logarithm.

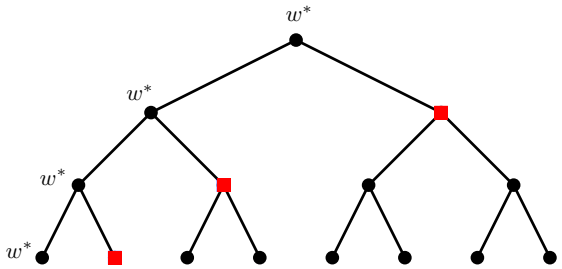

Figure 1: A tournament with $n = 8$ players where the winning player w^* is seeded into position one. The red vertices represent the roots of the three subtournaments that open.

If follows that

$$V^\star = \sum_{r=1}^{\log n} V_r^\star \leq \log n \cdot W_{\max}.$$

Overall, we have that

$$\frac{1}{\log n} V^\star \leq W_{\max} \leq V^\star,$$

and hence, the theorem follows. □

We remark that the bound presented in Theorem 7 is indeed tight for the presented algorithm. We illustrate this in the full version (Chaudhary, Molter, and Zehavi 2023).

A Quasi-Polynomial-Time Algorithm for Win-Count-Oriented Game-Value Functions. For TOURNAMENT VALUE MAXIMIZATION with a win-count-oriented game-value function (see Definition 2), we present a quasipolynomial-time algorithm.

Theorem 8. TOURNAMENT VALUE MAXIMIZATION *with a win-count-oriented game-value function can be solved in* $n^{\mathcal{O}(\log n)}$ *time.*

To describe the algorithm for Theorem 8, we introduce some additional terminology and concepts. Intuitively, we want to iterate through all players from the strongest to the weakest. We know that the strongest player wins the tournament, so we can assume w.l.o.g. that it is seeded into some fixed position, say one.

Once the winner of a tournament with n seed positions is placed in the seeding, we say that $\log n$ subtournaments *open up*, one for each round $r \in [\log n - 1]$ of the tournament except the last one and one degenerate subtournament with only one player (if $n > 1$, for "round zero"). The subtournament for round r has the 2^r seed positions $\{2^r + 1, 2^r + 2, \ldots, 2^{r+1}\}$. The degenerate subtournament for "round zero" has seed position 2. The winner of each of the subtournaments is seeded into the smallest seed position of that subtournament and loses against the winner of the overall tournament in the respective round. See Fig. 1 for an illustration.

For each subtournament, we now have an analogous situation. Since we seed players iteratively and decreasingly to their strength, we know that the first player seeded into a subtournament wins the subtournament. Once a winning player for a subtournament for some round r is seeded, we say that the subtournament is *closed*, and r new subtournaments open up, one for each of the $r - 1$ rounds of the subtournament for round r and one for "round zero".

Now, after a prefix of the players is seeded (according to their strength ordering), a certain set of subtournaments is open. We classify subtournaments by their size or, equivalently, by their number of rounds. We call a vector $\mathcal{X} = (x_0, x_1, \ldots, x_{\log n - 1}) \in \mathbb{N}^{\log n}$ a *subtournament profile*, where x_r quantifies the number of open subtournaments with r rounds.

We define the following dynamic program $T : \mathbb{N}^{\log n + 1} \to \mathbb{Z}$, where, intuitively, $T[\ell, \mathcal{X}]$ quantifies the maximum value achievable by seeding the ℓ strongest players while obtaining subtournament profile $\mathcal{X}$. Let $N = \{1, 2, \ldots, 2^{n'}\}$ be the set of players. Initialize T as follows.

$$T[1, x_0, x_1, \ldots, x_{n'-1}] =$$

$$\begin{cases} p(2^{n'}, n') & \text{if } x_0 = x_1 = \ldots = x_{n'-1} = 1, \\ -\infty & \text{otherwise.} \end{cases}$$

The remaining entries are recursively defined as follows.

$$T[\ell, x_0, x_1, \ldots, x_{n'-1}] =$$

$$\max_{r \in \{0,1,\ldots,n'-1\}} (T[\ell - 1, x_0 - 1, \ldots, x_{r-1} - 1,$$

$$x_r + 1, x_{r+1}, \ldots, x_{n'-1}] + p(2^{n'} - \ell + 1, r)).$$

We first prove the correctness of the dynamic program.

Lemma 9. $T[\ell, \mathcal{X}]$ *is the maximum value achievable by seeding the* ℓ *strongest players while obtaining subtournament profile* $\mathcal{X}$*.*

Proof. We prove the statement by induction on ℓ. Initially, for $\ell = 1$, we have $T[1, x_0, x_1, \ldots, x_{n'-1}] = p(2^{n'}, n')$ if $x_0 = x_1 = \ldots = x_{n'-1} = 1$, and $T[1, x_0, x_1, \ldots, x_{n'-1}] = -\infty$ otherwise.

Note that we start with the strongest player $i = 2^{n'}$. Hence, we have that i wins all games. It follows that for each round $r < n'$, one subtournament with r rounds is opened, and one degenerate tournament for round zero is opened. Hence, the only achievable subtournament profile is $\mathcal{X} = (1, 1, \ldots, 1)$ and the value achieved is $p(2^{n'}, n')$.

Now, assume that $\ell > 1$ and let $\mathcal{X} = (x_0, x_1, \ldots, x_{n'-1})$ be a subtournament profile. Note that when seeding a player as a winner of a subtournament with r rounds, one subtournament with r rounds is closed, and r subtournaments with $0, 1, \ldots, r - 1$ rounds, respectively, are opened. It follows that to achieve the subtournament profile $\mathcal{X} = (x_0, x_1, \ldots, x_{n'-1})$ by seeding the ℓth player as a winner of a subtournament with r rounds, the previous subtournament profile must be $\mathcal{X}' = (x_0 - 1, x_1 - 1, \ldots, x_{r-1} - 1, x_r + 1, x_{r+1}, x_{r+2}, \ldots, x_{n'-1})$. Seeding the ℓth strongest player as a winner of a subtournament with r rounds increases the tournament value by $p(2^{n'} - \ell + 1, r)$. By induction, we have that the optimal tournament value achievable by seeding player ℓ as a winner of a subtournament with r rounds and obtaining subtournament profile $\mathcal{X}$ is $T[\ell - 1, x_0 - 1, \ldots, x_{r-1} - 1, x_r + 1, x_{r+1}, \ldots, x_{n'-1}] +$

$p(2^{n'}-\ell+1, r)$. It follows that the optimal tournament value achievable by seeding the ℓ strongest players while obtaining subtournament profile $\mathcal{X}$ is the maximum over all the possibilities of how to seed the ℓth strongest player, which is $\max_{r\in\{0,1,\ldots,n'-1\}}(T[\ell-1, x_0-1, \ldots, x_{r-1}-1, x_r+1, x_{r+1}, \ldots, x_{n'-1}] + p(2^{n'}-\ell+1, r))$. It follows that the dynamic program is correct. □

Finally, we are ready to prove Theorem 8.

Proof of Theorem 8. Consider the dynamic programming table $T : \mathbb{N}^{\log n+1} \to \mathbb{N}$ described above. By Lemma 9, we have that we can find the optimal solution by computing $T[2^{n'}, 0, 0, \ldots, 0]$; when all players are seeded, all subtournaments are closed. Each entry of the table can be computed in time $\mathcal{O}(\log n)$ time, since we compute the maximum over $\mathcal{O}(\log n)$ values, each of which can be looked up in constant time. From the definition of the table, it follows that its size is in $n^{\mathcal{O}(\log n)}$, since we consider $\mathcal{O}(\log n)$ different sizes of subtournaments and $\mathcal{O}(n)$ subtournaments of each size can be open. Theorem 8 follows. □

Algorithms for Player Popularity-Based Game-Value Functions. We give a linear time greedy algorithm for special cases of player popularity-based game-value functions where there are two different player popularity values. Furthermore, we give an FPT algorithm for the "disagreement" between the player popularity values and the strength ordering, which, intuitively, measures how different the strength ordering is from the ordering obtained by sorting the players (descending) by their player popularity value. We give a formal definition when introducing the algorithms. The algorithms presented here use the concept of open and closed subtournaments introduced in the previous section.

Theorem 10 (⋆). TOURNAMENT VALUE MAXIMIZATION *is solvable in linear time if the game-value function is player popularity-based and there are only two different player popularity values.*

Proof. Let a and b denote the two distinct player popularity values with $a > b$. We call players with popularity value a *popular* and all other players *unpopular*.

We provide a simple greedy algorithm for this case that uses the concept of open and closed subtournaments. During the execution of the algorithm, we keep track of how many subtournaments of which size are currently open. Note that since we are given a tournament with n players, initially, one subtournament (with $\log n$ rounds) is open. Now, our algorithm performs the following steps.

1. The initial tournament value is set to 0, and one subtournament with $\log n$ rounds is open.
2. Starting with the strongest player, iterate through the players in order of their strength.
3. Let player i be the player considered in the current iteration. Let r denote the largest number of rounds of an open subtournament, and let r' denote the smallest number of rounds of an open subtournament.

3.1 If player i is popular:

3.1.1 Close one open subtournament with r rounds and open r new subtournaments as follows: for each round $r'' \in [r-1]$, we open one subtournament with r'' rounds. Furthermore, if $r \neq 0$, we open one degenerate subtournament for round zero.

3.1.2 Add $r \cdot a$ to the current tournament value.

3.2 If player i is unpopular:

3.2.1 Close one open subtournament with r' rounds and open r' new subtournaments as follows: for each round $r'' \in [r'-1]$, we open one subtournament with r'' rounds. Furthermore, if $r' \neq 0$, we open one degenerate subtournament for round zero.

3.2.2 Add $r' \cdot b$ to the current tournament value.

The running time analysis and the correctness proof of the described algorithm are deferred to the full version of this paper (Chaudhary, Molter, and Zehavi 2023). □

Next, we give an FPT algorithm for the case where the strength ordering of the players and the player popularity value ordering agree for most players. Formally, we say that the player popularity values *agree* with the strength ordering if for all $i, j \in N$ we have that if $i > j$ then $v_i \geq v_j$, where v_i and v_j are the player popularity values of i and j, respectively. We say that the *disagreement* between the player popularity values and the strength ordering is k if k is the smallest integer such that there exists a player set $N' \subseteq N$ with $|N'| \leq k$ and for all $i, j \in N \setminus N'$ we have that if $i > j$ then $v_i \geq v_j$, where v_i and v_j are the player popularity values of i and j, respectively. We call N' a minimum set of disagreeing players.

We use the disagreement between the player popularity values and the strength ordering as a parameter. Intuitively, we guess for the disagreeing players how many wins they should get, and then use the greedy algorithm of Theorem 10 treating the remaining players as if they were popular.

First, we show that we can compute a minimum set of disagreeing players efficiently, since we are going to need access to this set in our FPT algorithm.

Proposition 11 (⋆). *Given an instance of* TOURNAMENT VALUE MAXIMIZATION *with a player popularity-based game-value function, we can compute a minimum set of disagreeing players for that instance in $\mathcal{O}(k2^k \cdot n + n \log n)$ time, where k is the disagreement between the player popularity values and the strength ordering.*

Now, we are ready to present the FPT algorithm for the minimum disagreement as a parameter.

Theorem 12 (⋆). TOURNAMENT VALUE MAXIMIZATION *with a player popularity-based game-value function is solvable in $k^{\mathcal{O}(k)} \cdot n^{1+o(1)}$ time where k is the disagreement between player popularity values and the strength ordering.*

We point out that if $k = 0$, that is, the set of disagreeing players is empty, then we can solve the problem in linear time (rather than $n^{1+o(1)}$ time). We give more details on this in the full version (Chaudhary, Molter, and Zehavi 2023).

Corollary 13. TOURNAMENT VALUE MAXIMIZATION *with a player popularity-based game-value function is solvable in linear time if the player popularity values agree with the strength ordering*

We further remark that the corollary also follows from results obtained by Dagaev and Suzdaltsev (2018). Finally, we remark that TOURNAMENT VALUE MAXIMIZATION is presumably not NP-hard in the restricted setting, which our FPT algorithm is able to solve, since we have a quasi-polynomial time algorithm for this case (Theorem 8).

An FPT-Algorithm for Parameter Size of Influential Set of Players. In this section, we study the parameterized complexity of TOURNAMENT VALUE MAXIMIZATION when there exists a small *influential set of players*.

Definition 4. *Given an instance of* TOURNAMENT VALUE MAXIMIZATION*, a set $F \subseteq N$ is* influential *if $\forall\, i, j \in N$ and $r \in \mathbb{N}$ with $v(i, j, r) \neq 0$, we have that $i \in F$ or $j \in F$.*

First, we observe that we can compute a minimum-sized influential set of players efficiently, since we can easily reduce the problem to finding a minimum vertex cover in an appropriately defined graph.

Observation 14. *A minimum influential set F for an instance I of* TOURNAMENT VALUE MAXIMIZATION *can be computed in $1.2738^{|F|} \cdot |I|^{\mathcal{O}(1)}$ time.*

This follows from the fact that an influential set of players is a vertex cover of the graph that has the set of players N as vertices and an edge between two players $i, j \in N$ if there exists an $r \in \mathbb{N}$ such that $v(i, j, r) \neq 0$. VERTEX COVER is fixed-parameter tractable when parameterized by the solution size, and a minimum vertex cover can be computed in the claimed running time (Chen, Kanj, and Xia 2010).

Formally, we show the following.

Theorem 15 ($\star$)**.** TOURNAMENT VALUE MAXIMIZATION *with round-oblivious game-value function is fixed-parameter tractable when parameterized by the size of a minimum influential set of players.*

To show Theorem 15, we adapt an algorithm for TOURNAMENT FIXING parameterized by the feedback vertex number by Zehavi (2023). Due to the similarity of the algorithms, we need to introduce many concepts of Zehavi (2023). In TOURNAMENT FIXING, we are asked whether a seeding exists that makes a specific player win the tournament, given a set of players and a tournament graph that defines the outcome of a game between each pair of players. The parameter is the feedback vertex number of said tournament graph. Note that upon removing players corresponding to the feedback vertex set, the remaining players can be ordered linearly based on their strength, similar to our setting. Informally, the main strategy of the algorithm of Zehavi (2023) is as follows.

First, we "guess" a sufficient amount of information on how the players corresponding to the feedback vertex set perform in the tournament. Then, we find suitable opponents for the players corresponding to the feedback vertex set. Recall that those players do not fit into the linear ordering defining the strength. For each of those players, there are some games that they are supposed to win. For those games, we have to select players as opponents that they are able to beat. Similarly, for each of those players, there is at most one game that they are supposed to lose. For that game, we have to select a player as an opponent that beats them. Finally, we make sure that we can fill up the rest of the tournament with the remaining players.

We can solve our problem by following a similar approach. First, we want to "guess" a sufficient amount of information on how the players in the influential set of players perform in the tournament. Then, we want to find suitable opponents for the players in the influential set of players, since those games are the only ones giving us value. Finally, we want to make sure that we can fill up the rest of the tournament with the remaining players.

We defer the detailed description of the algorithm to the full version (Chaudhary, Molter, and Zehavi 2023).

5 Conclusion and Open Problems

In this paper, we formulated an objective of finding seedings for a knockout tournament that maximizes the profit or popularity of the tournaments. We provided a spectrum of results, showcasing both negative and (mostly) positive results. Still, several intriguing open questions remain.

Given that we have a quasipolynomial-time algorithm for TOURNAMENT VALUE MAXIMIZATION with a win-count-oriented game-value function, it is natural to ask the following: Is it possible to solve TOURNAMENT VALUE MAXIMIZATION for win-count-oriented game-value functions in polynomial time, or can we establish a conditional lower bound for the problem? What is the computational complexity of TOURNAMENT VALUE MAXIMIZATION when the game-value functions are player popularity-based and there are more than two distinct player popularity values?

We have restricted the game-value functions in our paper to be round oblivious extensively, as our primary aim was to lay a theoretical groundwork to fully understand and investigate the simpler game-value functions first. It seems very difficult, algorithmically, to handle game values that can arbitrarily change in different rounds unless all possibilities are checked, such as in our algorithm for win-count-oriented game value functions (Theorem 8), which can be non-round-oblivious. However, given the limitations of round obliviousness in real-world contexts, we believe that finding natural restrictions for non-round-oblivious game-value functions that still allow for additional algorithmic results is a promising future research direction.

Lastly, can we replace the parameter "size of the influential set of players" with a natural and smaller structural parameter and obtain tractability?

As the field continues to evolve, our research might inspire further exploration in optimizing tournament seeding strategies from the viewpoint of the organizers, including the investigation of probabilistic models.

Acknowledgements

J. Chaudhary and M. Zehavi are supported by the ERC, grant nr. 101039913; H. Molter by the ERC, grant nr. 949707.

References

Ausiello, G.; Crescenzi, P.; Gambosi, G.; Kann, V.; Marchetti-Spaccamela, A.; and Protasi, M. 2012. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media.

Aziz, H.; Gaspers, S.; Mackenzie, S.; Mattei, N.; Stursberg, P.; and Walsh, T. 2018. Fixing balanced knockout and double elimination tournaments. *Artificial Intelligence*, 262: 1–14.

Chaudhary, J.; Molter, H.; and Zehavi, M. 2023. How to Make Knockout Tournaments More Popular? *CoRR*, abs/2309.09967.

Chen, J.; Kanj, I. A.; and Xia, G. 2010. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40-42): 3736–3756.

Chmait, N.; Westerbeek, H.; Eime, R.; Robertson, S.; Sellitto, C.; and Reid, M. 2020. Tennis influencers: The player effect on social media engagement and demand for tournament attendance. *Telematics and Informatics*, 50: 101381.

Connolly; and Rendleman. 2011. Tournament qualification, seeding and selection efficiency. *Technical Report 2011-96, Tuck School of Business*.

Dagaev, D.; and Suzdaltsev, A. 2018. Competitive intensity and quality maximizing seedings in knock-out tournaments. *Journal of Combinatorial Optimization*, 35: 170–188.

ESPN. 2023a. Men's Tennis ATP Rankings 2023. https://www.espn.com/tennis/rankings. Accessed: 2023-12-19.

ESPN. 2023b. Women's Tennis ATP Rankings 2023. https://www.espn.com/tennis/rankings/_/type/wta. Accessed: 2023-12-19.

Galil, Z. 1986. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1): 23–38.

Groh; Moldovanu; Sela; and Sunde. 2012. Optimal seedings in elimination tournaments. *Economic Theory*, 49(1): 59–80.

Gupta, S.; Roy, S.; Saurabh, S.; and Zehavi, M. 2018a. When Rigging a Tournament, Let Greediness Blind You. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 275–281.

Gupta, S.; Roy, S.; Saurabh, S.; and Zehavi, M. 2018b. Winning a Tournament by Any Means Necessary. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 282–288.

Gupta, S.; Saurabh, S.; Sridharan, R.; and Zehavi, M. 2019. On Succinct Encodings for the Tournament Fixing Problem. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 322–328.

Hall, J.; O'Mahony, B.; and Vieceli, J. 2010. An empirical model of attendance factors at major sporting events. *International journal of hospitality management*, 29(2): 328–334.

Horen; and Riezman. 1985. Comparing Draws for Single Elimination Tournaments. *Operations Research*, 33(2): 249–262.

Kim, M. P.; Suksompong, W.; and Williams, V. V. 2017. Who can win a single-elimination tournament? *SIAM Journal on Discrete Mathematics*, 31(3): 1751–1764.

Kim, M. P.; and Williams, V. V. 2015. Fixing Tournaments for Kings, Chokers, and More. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 561–567.

Laslier. 1997. Tournament Solutions and Majority Voting. *Springer-Verlag*.

Manurangsi, P.; and Suksompong, W. 2023. Fixing knockout tournaments with seeds. *Discrete Applied Mathematics*, 339: 21–35.

Ramanujan, M. S.; and Szeider, S. 2017. Rigging Nearly Acyclic Tournaments Is Fixed-Parameter Tractable. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 3929–3935.

Rosen. 1986. Prizes and incentives in elimination tournaments. *The American Economic Review*, 76(4): 701–715.

Stanton, I.; and Williams, V. V. 2011a. Manipulating Stochastically Generated Single-Elimination Tournaments for Nearly All Players. In *Proceedings of the 7th International Workshop on Internet and Network Economics (WINE)*, 326–337.

Stanton, I.; and Williams, V. V. 2011b. Rigging Tournament Brackets for Weaker Players. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 357–364.

Suksompong, W. 2021. Tournaments in Computational Social Choice: Recent Developments. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 4611–4618.

Tullock. 1980. Toward a Theory of the Rent-seeking Society. *Texas A&M University Press*.

Vu, T.; Altman, A.; and Shoham, Y. 2009. On the complexity of schedule control problems for knockout tournaments. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 225–232.

Williams, V. V. 2010. Fixing a Tournament. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*.

Williams, V. V.; and Moulin, H. 2016. *Knockout Tournaments*, 453–474. Cambridge University Press.

Zehavi, M. 2023. Tournament Fixing Parameterized by Feedback Vertex Set Number Is FPT. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, volume 37:5, 5876–5883.