Maximizing Value in Challenge the Champ Tournaments

Umang Bhaskar
Tata Institute of Fundamental Research, India
umang@tifr.res.in

Juhi Chaudhary
Tata Institute of Fundamental Research, India
juhi.chaudhary@tifr.res.in

Palash Dey
IIT Kharagpur, India
palash.dey@cse.iitkgp.ac.in

ABSTRACT

A tournament is a method to decide the winner in a competition, and describes the overall sequence in which matches between the players are held. While deciding a worthy winner is the primary goal of a tournament, a close second is to maximize the value generated for the matches played, with value for a match measured either in terms of tickets sold, television viewership, advertising revenue, or other means. Tournament organizers often seed the players — i.e., decide which matches are played — to increase this value.

We study the value maximization objective in a particular tournament format called *Challenge the Champ*. This is a simple tournament format where an ordering of the players is decided. The first player in this order is the initial champion. The remaining players in order challenge the current champion; if a challenger wins, she replaces the current champion. We model the outcome of a match between two players using a complete directed graph, called a *strength graph*, with each player represented as a vertex, and the direction of an edge indicating the winner in a match. The value-maximization objective has been recently explored for knockout tournaments when the strength graph is a directed acyclic graph (DAG). We extend the investigation to Challenge the Champ tournaments and general strength graphs. We study different representations of the value of each match, and completely characterize the computational complexity of the problem.

KEYWORDS

Tournaments; Challenge the Champ; Algorithms; NP-hardness; Value Maximization

ACM Reference Format:
Umang Bhaskar, Juhi Chaudhary, and Palash Dey. 2025. Maximizing Value in Challenge the Champ Tournaments. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Tournaments are a fundamental mechanism for determining winners in diverse competitive settings by systematically comparing participants through a series of pairwise matches.[1] The most illustrative examples of tournaments are sports competitions, ranging from prestigious international events like the Olympics and World Cups to local school sports and college contests. Different tournament formats, such as *knockout* tournament [7, 22, 25], *round-robin* tournament [18, 21], *double-elimination tournament* [2, 20], *Swiss-Sytem* [19, 23] are commonly employed and have been widely studied by researchers over the years, where each format offers a distinct approach with its own advantages and disadvantages for ranking and elimination.

Challenge the Champ — also known as stepladder — is one such tournament format, which, despite its simplicity, is relatively underexplored in the literature compared to other tournament variants. Given n players, a Challenge the Champ tournament proceeds in $n-1$ rounds. A player is initially chosen as the champ, and in each successive round, a new player challenges the current champ. If a player defeats the current champ, she becomes the new champ. The champ at the end of the final round is the tournament winner. See Figure 1 for an illustration. Challenge the Champ is a *single-elimination tournament*, where a player is eliminated once she loses a match. Thus, in a single-elimination tournament with n players, there are exactly $n-1$ matches, though there are of course many ways to decide the players in each match.[2]

Note that the order in which the players challenge the current champ — called the *seeding* — significantly influences the outcome of the tournament. E.g., a fixed player i has a better likelihood of winning the tournament if she challenges the champ late in the tournament, after the stronger players have been eliminated, rather than very early.

Owing to their simplicity and often dramatic nature — allowing a new entrant to beat the current champion and take over the title — Challenge the Champ tournaments and its variants are used in multiple sports, including ten-pin bowling, squash, badminton, and basketball [1, 24]. Beyond the world of sports, a continuous variant of stepladder tournaments is also used for ranking workers in organisations [14], making it a compelling tournament format for theoretical studies. In fact, some of our results obtained from studying Challenge the Champ tournaments are applicable to all single-elimination tournaments, further reinforcing their importance.

Challenge the Champ tournaments have previously attracted some attention from researchers, though this has been focused on the complexity of manipulation to ensure a given player wins the tournament [6, 13]. We are, however, interested in the objective of value maximization in sports tournaments. Arguably, many competitions are organized to maximize some measure of value — whether advertising revenue, viewership, or attendance. Given a tournament format — such as knockout, or Swiss-system — often this value-maximization objective influences the matches that are

[1] In this paper, a *tournament* will denote a competition format, and not a complete directed graph.

[2] A *knockout* tournament, which takes the form of a binary tree, is also sometimes referred to as a single-elimination tournament. However, we use the more general definition of single-elimination tournaments in this paper.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Table 1: Our results for value maximization in Challenge the Champ tournaments.

Problem	Pair-Based	Win-Count-Based	Player-Popularity-Based
CTC-VM-DAG	NP-complete for binary values (Theorem 11)	P (Theorem 7)	P (Theorem 2)
CTC-VM	NP-complete (Follows from Theorem 11)	NP-complete for binary values (Theorem 8) and linear-after-threshold (Theorem 10)	P for binary values (Theorem 3) NP-complete for ternary values (Theorem 5)

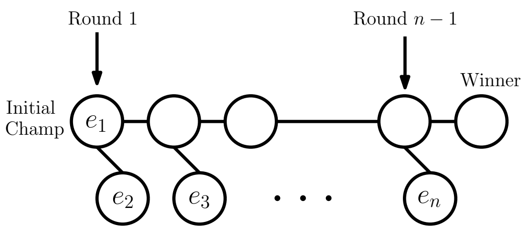

Figure 1: A Challenge the Champ tournament, with players seeded $(e_1, e_2, \ldots, e_n)$. In each round i, player e_{i+1} challenges the current champ — the winner of the previous round.

played as well, e.g., through the choice of groups, or seeding of the players. This objective is measured by assigning a non-negative integer value to each potential match between the players. Such an approach reflects the practical scenario where matches hold varying levels of importance, influenced by factors like geopolitics, historical rivalries, intrinsic fan following, or the relative strength of the teams. Prioritizing high-value matches becomes essential, especially when only a subset of potential matches can be played, as it helps attract viewers and advertisers. The *tournament value* is then defined as the total sum of the values of all matches that are actually played. This metric is crucial for evaluating and optimizing the overall appeal and financial success of the tournament. We study the problem of value maximization in Challenge the Champ tournaments. Given the expanding body of research on tournaments, we believe that studying value maximization in the simple but interesting Challenge the Champ tournament format is an important research problem.

To complete the picture, we need a way of determining the outcome when two players compete. For this, we use a *strength graph*, which is a complete directed graph with a vertex for each player. A directed edge (i, j) indicates that player i beats player j in a match. Although value maximization has been previously examined in knockout tournaments [5, 10], the investigation is restricted to the case where the strength graph is a directed acyclic graph (DAG). Our study extends this inquiry to Challenge the Champ tournaments. We investigate scenarios where the strength graph is either a DAG or a complete directed graph, and provide a thorough analysis of the computational complexity of value maximization in Challenge the Champ tournaments.

Related Work

Challenge the Champ tournaments are also sometimes called stepladder tournaments in previous work. These have been studied from the perspective of satisfying axiomatic notions of fairness [1], as well as the characterizing strength graphs where a favorite player can win the tournament [4, 26]. Arlegi and Dimitrov [1] highlight instances of stepladder tournaments being utilized in sports competitions, such as ten-pin bowling and squash. Stepladder tournaments are also useful in firms. Pongou et al. [14] present examples, and discuss the relation of the ranking of workers obtained from stepladder tournaments with their importance in the firm.

For Challenge the Champ tournaments with probabilistic strength graphs (rather than deterministic, as in our case), Mattei et al. [13] investigated a setting where players can be bribed to lower their winning probability against the initial champ. The goal was to maximize the probability of the initial champ winning the tournament by bribing the other players while not exceeding a given budget for the bribes. Building on this, Chaudhary et al. [6] extended the research by examining the problem with respect to various parameters.

For tournament value maximization, Chaudhary et al. [5] initiated this study for knockout tournaments under the name TOURNAMENT VALUE MAXIMIZATION in a deterministic setting, where players have a strict strength ordering (the strength graph forms a DAG). Their work explored various constraints on tournament value functions to optimize the overall tournament value. In a related study, Gupta et al. [10] examined a similar problem but without assuming a strict linear order of player strengths. Their focus was on determining whether a specific seeding could guarantee that a designated set of games, known as demand matches, would occur. This problem can be viewed as a special case of TOURNAMENT VALUE MAXIMIZATION, where demand matches are assigned a value of 1, and other matches a value of 0, with the objective of ensuring the total value meets or exceeds the number of demand matches. Additionally, Dagaev and Suzdaltsev [8] investigated a restricted variant of tournament value maximization, where each player has a unique strength value, and the value of a game is defined by a linear combination of its "quality" (the sum of the players' strengths) and its "intensity" (the absolute difference between the players' strengths). They characterized scenarios in which either a "close" seeding, a "distant" seeding, or any seeding could be optimal, demonstrating that their restricted cases can be solved efficiently in linear time.

2 PROBLEM STATEMENT AND PRELIMINARIES

For a positive integer n, define $[n] := \{1, 2, \ldots, n\}$. An instance of CHALLENGE THE CHAMP VALUE MAXIMIZATION is given by a set of players $N = [n]$, a strength graph, which is a complete directed graph $\mathcal{T} = (N, E)$ with a directed edge (i, j) indicating that player

i beats player j in a match, and a value function v for each possible match in the tournament. We present results for both the case when $\mathcal{T}$ is a DAG and when $\mathcal{T}$ contains directed cycles. If $\mathcal{T}$ is a DAG, we will assume all edges are oriented from larger indices to smaller indices. Hence n beats every other player, while 1 is beaten by every other player. If a player i defeats a player j, we say that i is *stronger* than j, or equivalently, j is *weaker* than i.

Given a Challenge the Champ Value Maximization instance with n players, a seeding $\sigma \in \mathbb{S}_n$ (where $\mathbb{S}_n$ is the symmetric group on n elements) is a permutation of the set of players that completely determines the order of the matches. The player $\sigma(1)$ is the initial champ. In each round $r = 1, \ldots, n-1$, the champ is challenged by player $\sigma(r+1)$. The winner of the match (as determined by the strength graph $\mathcal{T}$) is the champ for the next time step. The winner in the last round is the winner of the tournament. Thus, given N and $\mathcal{T}$, each seeding σ completely determines the set of matches that take place. Given a seeding σ, we define $M_i(\sigma)$ as the set of players that play a match with player i and lose to it, and define $w_i(\sigma) = |M_i(\sigma)|$. Note that player i wins exactly $w_i(\sigma)$ matches.

We consider a number of different possible settings for the tournament value $V(\sigma)$ of a seeding σ for a Challenge the Champ tournament.

Player-popularity-based: We first consider a natural restriction when each player has an associated popularity $p_i \in \mathbb{Z}_+$, and the value of each match is determined by the popularity of the winning player. Thus, each match player i wins contributes value p_i. Thus the value for the seeding σ is $V(\sigma) = \sum_{i \in N} p_i w_i(\sigma)$, and such a value function is called *player-popularity-based.* This restriction captures the intrinsic popularity of certain teams and players among their fans, irrespective of the opponent or venue. This variant has also been explored by Chaudhary et al. [5].

Mathematically, when the strength graph is a DAG, the player-popularity-based value function is *equivalent* to the tournament value function where the value of every match is defined as the sum of the popularity values of the two players participating in that match. This equivalence holds for any single elimination tournament (not necessarily Challenge the Champ), if the strength graph is a DAG. The equivalence can be seen easily: Let $V'(\sigma)$ denote the tournament value when the value of the match between players i and j is $v'(i,j) = p_i + p_j$, and let $V(\sigma)$ denote the tournament value when the value of the match is $v(i,j) = p_i$, where $i > j$. Then $V'(\sigma) = V(\sigma) + \sum_{i \in [n-1]} p_i$, since every player except player n loses exactly one match in any single elimination tournament. Thus, when the match values are modified, all players except the winner additionally contribute their popularity value exactly once, for the match where they lose. Thus, since the two tournament value functions are equivalent for DAGs, we will present results for the player-popularity-based value function.

Win-count-based: We next consider tournament value functions where the value of each match depends not only on the winning player, but also by her track record of victories. Specifically, the value of a match increases with the number of wins the winning player has accumulated. We refer to such value functions as *win-count-based* value functions. Formally, each player has a win-value $f_i : [n-1] \rightarrow \mathbb{Z}_+$ that gives the value of each match won by player i. I.e., the kth match won by player i has value $f_i(k)$. Then the value for the seeding σ for win-count-based value function is $V(\sigma) = \sum_{i \in N} \sum_{k=1}^{w_i(\sigma)} f_i(k)$. Note that if each player has a constant win-value function , i.e., if $f_i(k) = p_i$ for every $k \in [w_i(\sigma)]$, this is the same as player-popularity-based value functions.

Within win-count-based values, we consider two further restrictions. We say a win-count-based tournament value function is *binary-valued* if every player i has a threshold λ_i, and the win-value function $f_i(x) = 1$ if $x = \lambda_i$, and is zero otherwise. Thus, player i gets a value of 1 if she wins at least λ_i games, and gets value 0 otherwise. The tournament value function is thus $V(\sigma) = |\{i : w_i(\sigma) \geqslant \lambda_i\}|$. These value functions support the egalitarian objective of ensuring that more players/teams achieve their individual thresholds. Maximizing the tournament value, in this case, is the same as obtaining a seeding that enables as many players as possible to meet their thresholds, rather than allowing a single player to dominate by participating in the majority of matches.

We say a win-count-based tournament value function is *linear-after-threshold* if every player i has a threshold λ_i, and the win-value function $f_i(x) = 1$ if $x \geqslant \lambda_i$, and is zero otherwise. Thus, player i gets a value of 1 for every game she wins after $\lambda_i - 1$, and the tournament value function is $V(\sigma) = \sum_{i \in N} \max\{0, w_i(\sigma) - \lambda_i + 1\}$.

Pair-based: Finally, we consider *pair-based* value function, where the value of each match depends on both the players in the match. Thus, such value functions account for matches where the players have a historic rivalry or are particularly competitive. Here, each pair of players i, j has a pair value $f_{ij} \in \mathbb{Z}_+$, which is the value contributed by a match between i and j. The value of a seeding σ is $V(\sigma) = \sum_{i \in N} \sum_{j \in M_i(\sigma)} f_{ij}$.

Again, pair-based value functions generalize player-popularity-based value functions. To see this, if each player i has popularity p_i, consider the pair-based value function where the pair $\{i, j\}$ has value p_i if i beats j, and p_j otherwise. Clearly, each match then has the same value in either case.

These value functions have been studied before for value maximization in knockout tournaments [5]. Pair-based value functions are also called *round-oblivious* value functions. We will further consider restrictions on the value of each match to be binary or ternary (i.e., the value of each match is in $\{0, 1\}$ or in $\{0, 1, 2\}$).

More formally, the definition of Challenge the Champ Value Maximization is given below.

Challenge the Champ Value Maximization (CTC-VM)	
Input:	Given a set N of players, a strength graph $\mathcal{T}$ on N which is a complete directed graph, a tournament value function V, and a target value t.
Question:	Is there a tournament seeding σ for the players in N such that the tournament value $V(\sigma) \geqslant t$?

The problem Challenge the Champ Value Maximization-Dag (CTC-VM-Dag) is defined in a similar manner, except that the input strength graph, in this case, is a DAG.

Seedings and Caterpillars. For undirected graphs, a *caterpillar graph* is a tree in which the removal of all pendant vertices (leaves) yields a path, which we refer to as the *backbone*. We extend this to directed graphs: a *caterpillar arborescence* is an arborescence where removing all leaves (vertices with out-degree zero) yields a directed path. We call this path the *backbone* of the caterpillar. A *spanning caterpillar arborescence* of a directed graph is a subgraph that is a caterpillar arborescence on all the vertices. For brevity, since we will only consider spanning caterpillar arborescences on the directed graph $\mathcal{T}$, we omit the term arborescence from the description.

Any seeding σ in a Challenge the Champ tournament induces a spanning caterpillar in the strength graph, and vice versa. To see this, given a seeding σ, let the backbone of the caterpillar be the players that win at least one match, in the order given by σ. All remaining players lose their first match to a player in the backbone. Attach the remaining players to the player they lose to in the backbone. This gives a spanning caterpillar. For the converse, given a spanning caterpillar, the seeding is obtained by the sequence of players in the backbone, interspersed by the leaves attached to each player in the backbone in any order. Figure 2 gives an example showing this correspondence. Thus, instead of a seeding, we can equivalently specify the sequence of players in the backbone of a caterpillar, and the attachment of the remaining players (i.e., the leaves) to the players in the backbone. We will frequently use this correspondence in our proofs. Corollary 12 also establishes a connection between pair-based value functions and spanning caterpillars. We note that there is prior work linking spanning caterpillars and tournaments (see, e.g., [16, 17]). Our work significantly strengthens this connection by demonstrating that caterpillars provide a natural framework for understanding Challenge the Champ tournaments and establishing hardness results for caterpillars.

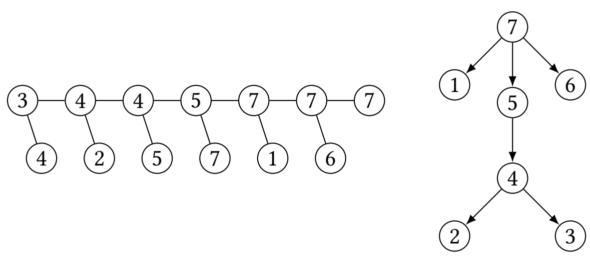

Figure 2: The first figure gives an example with 7 players showing the seeding $(3, 4, 2, 5, 7, 1, 6)$. In the strength graph (not shown), each player j defeats all players $i < j$. Hence player 7 is the strongest and player 1 the weakest. The second figure shows the resulting spanning caterpillar arborescence.

We will also use the following result in our proofs.

Proposition 1 ([15]). *Given a complete directed graph G, a Hamiltonian path in G can be found in polynomial time.*

Our Contribution

We present a comprehensive analysis of the complexity of tournament value maximization for Challenge the Champ tournaments. Our work encompasses both CTC-VM and CTC-VM-Dag across various tournament value functions, including player-popularity-based, win-count-based, and pair-based tournament value functions, and restrictions to binary and ternary-valued functions. For a tabular view of the main results, see Table 1.

We start by delving into player-popularity-based tournament value functions in Section 3. We first establish that when the strength graph is a DAG, the tournament value can be maximized in polynomial time. Further, in this setting, Challenge the Champ tournaments are optimal among all single-elimination tournaments. Specifically, we show:

- For every strength graph with player-popularity-based tournament value functions, there exists a Challenge the Champ tournament that maximizes the total value over all single-elimination tournaments. Moreover, this maximum value and the seeding can be computed in polynomial time.

We note that in knockout tournaments, this problem is unresolved [5]. Our proof, however, shows that the optimal value obtained for Challenge the Champ tournaments is an upper bound for all single-elimination tournaments, including knockout tournaments.

When the strength graph is no longer a DAG, the complexity of the problem varies with the number of distinct player popularity values. For binary values for the popularity of players, we give a polynomial-time algorithm, and observe that in this setting again, Challenge the Champ tournaments are optimal among all single-elimination tournaments. However, when the popularity of the players can take three values $\{0, 1, 2\}$, the problem becomes NP-hard, *even when there is a single player with popularity 2,* which implies that the problem is para-NP-hard when parameterized by the number of players with popularity value 2.

- CTC-VM is polynomial-time solvable for player-popularity-based tournament value functions when each player's popularity is in $\{0, 1\}$.
- CTC-VM is NP-complete for player-popularity-based tournament value functions when each player's popularity is in $\{0, 1, 2\}$.

We also present a simple greedy approximation algorithm for CTC-VM, where each player's popularity is drawn from a set of k distinct values by leveraging the result of CTC-VM for player-popularity-based tournament value functions where each player's popularity is restricted to $\{0, 1\}$.

In Section 4, we explore the more general case of win-count-based tournament value functions. We provide a polynomial-time dynamic programming algorithm for the case where the input is a DAG. However, for general strength graphs, the problem becomes hard even if the win-value function f_i for each player is binary. This highlights that the presence of directed cycles is a crucial factor in the complexity of the problem.

- CTC-VM-Dag is polynomial-time solvable for win-count-based tournament value functions.
- CTC-VM is NP-complete even for binary-valued and linear-after-threshold win-count-based tournament value functions.

The last result also highlights a difference from player-popularity-based functions: while the latter becomes complex with ternary values, the former is challenging even with binary values.

In Section 5, we investigate pair-based tournament value functions and prove that the problem is NP-complete even for the simplest setting when the strength graph is a DAG, and each pair value is binary. Specifically, we prove:

- CTC-VM-Dag is NP-complete for pair-based tournament value functions when all pair values are binary.

We find this surprising, given the relative simplicity of Challenge the Champ tournaments compared to other tournament formats. Finally, we show that this last result also implies hardness for a problem of value maximization for spanning caterpillars.

3 PLAYER-POPULARITY-BASED TOURNAMENT VALUE FUNCTIONS

We first demonstrate that for player-popularity-based tournament value functions with acyclic strength graphs, the optimal tournament value can be obtained in polynomial time.

Theorem 2. *There is a polynomial-time algorithm for CTC-VM-Dag for player-popularity-based tournament value functions.*

Proof. As stated, since $\mathcal{T}$ is a DAG, we assume player $i \in N$ beats all players $j < i$, and player i has popularity p_i. We now describe a greedy algorithm that constructs an optimal spanning caterpillar. We assume $p_i \neq p_j$ for any two players i, j, else we can slightly perturb the popularity values. Now let player i_1 be the most popular player. Let $W_1 := [i_1 - 1]$ be all the players weaker than i_1. Let player i_2 be the most popular player in $N \setminus [i_1]$, and let $W_2 := [i_2 - 1] \setminus [i_1]$. Thus recursively, we define i_j to be the most popular player in $N \setminus [i_{j-1}]$, and $W_j := [i_j - 1] \setminus [i_{j-1}]$. We continue until we pick player n, i.e., for some k, $i_k = n$. Note that each player in W_j is less popular and weaker than player i_j, and player i_j is stronger than player i_{j-1}.

The spanning caterpillar has players $i_k, i_{k-1}, \ldots, i_1$ (in this order) on the backbone, and each i_j additionally has edges to all the players in W_j. Each player i_j in the backbone (other than i_1) thus wins $|W_j| + 1$ matches, while player i_1 wins $|W_1|$ matches. The total value is thus

$$(i_1 - 1) \cdot p_{i_1} + (i_2 - i_1) \cdot p_{i_2} + \ldots + (n - i_{k-1}) \cdot p_n.$$

We now show that this is, in fact, the maximum value obtainable in any seeding. To see this, consider any other seeding σ. Then there are $n - 1$ matches played, and there are $n - 1$ values obtained from these matches. Consider the tth largest value obtained, say v^*, for any $t \in [n - 1]$. We will show that this value is at most the tth largest value obtained by our algorithm, completing the proof.

Let $j \in [k]$ be the largest index so that $t \geqslant i_j$. Note that i_{j+1} is the most popular player in $N \setminus [i_j]$. Then it can be checked that the tth largest value obtained by our algorithm is $p_{i_{j+1}}$. Now assume for a contradiction that $v^* > p_{i_{j+1}}$. Since v^* is the tth largest value in the seeding σ, players with popularity at least v^* must have won at least t matches in total. But all players with popularity greater than i_{j+1} lie in $[i_j]$. Hence these players can win at most $i_j - 1$, which is strictly less than t. This gives a contradiction. □

As noted, the proof of Theorem 2 shows something stronger. Any single-elimination tournament consists of at most $n - 1$ matches, since in every match, one player is eliminated. Our proof shows that if the strength graph is a DAG, and the value of every match is the popularity of the stronger player, then the Challenge the Champ tournament, obtains maximum value over all possible single-elimination tournaments, and this value (and the seeding that obtains it) can be computed in polynomial time. To see this, it is enough to consider in the proof the tth largest value obtained from any single-elimination tournament; the proof shows that this is at most the value obtained by the seeding given for the Challenge the Champ tournament.

However, this is not the case when the strength graph is not necessarily a DAG. Even for player-popularity-based tournament value functions, there are instances where different tournament trees can be optimal. Figure 3 gives such an example. For *binary* player-popularity values, however, Challenge the Champ tournaments can again be shown to be optimal for arbitrary strength graphs among all single-elimination tournaments.

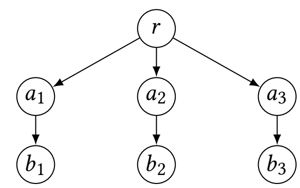

Figure 3: An example of a (partial) strength graph with cycles where a Challenge the Champ tournament is not optimal. Each leaf b_i beats all the non-leaf nodes except a_i, creating cycles. The root has popularity 2, intermediate a_i vertices have popularity 1, and the leaves have popularity 0. Here, the maximum value achievable by a single elimination tournament is 9 whereas the maximum value achievable in a Challenge the Champ tournament is 7.

Next, we show that CTC-VM can be solved in polynomial time for player-popularity-based tournament value functions when each player's popularity is either 0 or 1. This result is significant because here, as for DAGs, not only do we get a polynomial time algorithm, but the value obtained is the largest among all single-elimination tournaments. Further, for general strength graphs, this is the largest class of valuations for which we obtain positive complexity results. For ternary popularity values, the problem becomes NP-hard, as we show later.

Theorem 3. *There is a polynomial-time algorithm for CTC-VM for player-popularity-based tournament value functions when each player's popularity is either 0 or 1. The value obtained is the maximum among all single-elimination tournaments.*

Proof. Let $(N, \mathcal{T}, V, t)$ be a given instance of CTC-VM. Since the popularity value of every player maps only to $\{0, 1\}$, we categorize players with a popularity value of 1 as *popular* and those with a popularity value of 0 as *unpopular*. Next, we partition the players in N into three sets as follows:

- Popular players, denoted by $\mathcal{P}$.
- Unpopular players who defeat all popular players, denoted by $\mathcal{W}$.
- Unpopular players who are defeated by at least one popular player, denoted by $\mathcal{U}$.

First, we show that the maximum achievable tournament value is $|\mathcal{P}| + |\mathcal{U}| - 1$, and this in fact holds for all single-elimination tournaments. Second, we provide a seeding strategy that achieves this maximum value.

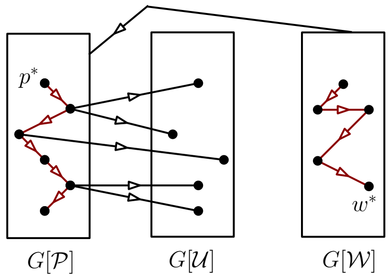

Figure 4: The red paths illustrate the Hamiltonian paths. A directed edge between two boxes represents all directed edges connecting the vertices from one box to those in the other.

For the first part, note that in any single-elimination tournament, matches involving players in $\mathcal{W}$ do not contribute to the tournament value, no matter who wins or loses. Matches where both players are in $\mathcal{U}$ similarly contribute nothing. A match between a player in $\mathcal{U}$ and a player in $\mathcal{P}$ contributes 1 if the player in $\mathcal{U}$ loses, hence these matches contribute at most $|\mathcal{U}|$. Matches between players in $\mathcal{P}$ can contribute a maximum of $|\mathcal{P}| - 1$ to the total tournament value. Consequently, the tournament value is bounded by $|\mathcal{P}| + |\mathcal{U}| - 1$.

For the second part, note that by Proposition 1, there exists a Hamiltonian path in both $G[\mathcal{P}]$[3] and $G[\mathcal{W}]$. Let P denote such a Hamiltonian path in $G[\mathcal{P}]$, and let p^* be the starting node of P. Let W denote a Hamiltonian path in $G[\mathcal{W}]$, and let w^* be the ending node of W. See Figure 4 for an illustration. Now, consider the Hamiltonian paths W and P as forming the backbone of a spanning caterpillar. This arrangement is valid since w^* beats p^* by definition of $\mathcal{W}$. Next, incorporate the vertices in $\mathcal{U}$ by attaching them as leaves to the vertex in $\mathcal{P}$ that beats them in the backbone. The resulting graph forms a spanning caterpillar with a tournament value of $|\mathcal{P}| + |\mathcal{U}| - 1$. □

Next, we prove that CTC-VM is NP-hard when each player's popularity lies in $\{0, 1, 2\}$. Surprisingly, our hardness result holds even if there is a single player with popularity value 2. This gives a very precise threshold, as we have seen that for binary popularity values, the tournament value can be maximized in polynomial time.

We give a reduction from 3-D-Matching. The input of 3-D-Matching consists of a finite set $\mathcal{U} = X \cup Y \cup Z$ where $|X| = |Y| = |Z| = n$, and a collection $\mathcal{S} = \{S_1, \dots, S_m\}$ of triples, where each $S_i \subseteq X \times Y \times Z$. The problem is to determine if there exists a subset $\mathcal{S}' \subseteq \mathcal{S}$ consisting of n triples such that each element in $\mathcal{U}$ appears in exactly one triple in $\mathcal{S}'$ [9].

We will use the following proposition for 3-D-Matching.

Proposition 4. *For an instance of* 3-D-Matching, *if a family of subsets $\mathcal{S}' \subseteq \mathcal{S}$ covers elements $\mathcal{U}' \subseteq \mathcal{U}$, then $|\mathcal{U}'| - |\mathcal{S}'| \leqslant 2n$, with equality iff $|\mathcal{U}'| = 3n$ and $|\mathcal{S}'| = n$. Hence if the given instance is a NO instance, then the inequality is always strict.*

Proof. Let $k = |\mathcal{S}'|$ and $\ell = |\mathcal{U}'|$. If $k < n$, then $\ell \leqslant 3k$ since each set contains 3 elements, and $\ell - k \leqslant 2k < 2n$. If $k > n$, then $\ell \leqslant 3n$, and again $\ell - k \leqslant 3n - k < 2n$. Thus if $\ell - k = 2n$, then $k = n$ and $\ell = 3n$. □

Theorem 5. *CTC-VM is NP-complete for player-popularity-based tournament value functions when each player's popularity is either 0, 1, or 2.*

Proof. The reduction from 3-D-Matching is as follows. We create a strength graph $\mathcal{T}$ with vertex set N. Let H, M, and L denote a partition of N that corresponds to players with popularity 2, 1, and 0, respectively. Corresponding to every element $a \in \mathcal{U}$, introduce a player v_a in L. For each set $S \in \mathcal{S}$, introduce a player v_S in M. If $a \in S$ for some $a \in \mathcal{U}$ and $S \in \mathcal{S}$, then the edge between v_S and v_a is directed from v_S to v_a, else it is directed from v_a to v_S. We introduce a new player h in H that beats every player in M and loses to every player in L. The remaining edges in $\mathcal{T}$ are directed arbitrarily. We set the target value t as $2(|\mathcal{S}| - n) + 3n + 2 + (n - 1) = 2|\mathcal{S}| + 2n + 1$. The intuition behind the target value is that since each player in M must choose between contributing to the tournament value by losing to h (and thereby adding 2 points to the tournament value) or by defeating three players in L. Note that the factor $n - 1$ arises because the players in M who are part of the backbone can contribute additional points by playing among themselves. Also, the extra 2 is due to the fact that, at most, one player in the backbone that comes from M can contribute to the tournament value by both beating some players and losing to h.

We need to show that $(\mathcal{U}, \mathcal{S})$ is a YES-instance of 3-D-Matching iff $(N, \mathcal{T}, V, t)$ is a YES-instance of CTC-VM.

For the forward direction, let $\mathcal{S}'$ denote the subset of $\mathcal{S}$ of size n such that each element in $\mathcal{U}$ appears in exactly one triple in $\mathcal{S}'$. Now we will construct a spanning caterpillar whose tournament value is at least $2|\mathcal{S}| + 2n + 1$ as follows. Let h belong to the backbone. For the sets in $\mathcal{S}'$, let $M' \subseteq M$ denote the corresponding players. These players will be part of the backbone of the spanning caterpillar. Using Proposition 1, we can find a Hamiltonian path in the subgraph $G[M']$. We attach this Hamiltonian path after the player h in the backbone of the spanning caterpillar. This step is valid because player h beats every player in M. Players in L are attached to the player in M' in the backbone that beats them (recall that v_S beats v_a iff $a \in S$). Players in M corresponding to sets not in $\mathcal{S}'$ are attached to player h as leaves. Note that the value of the tournament is $2|\mathcal{S}| + 2n + 1$.

For the reverse direction, assume the tournament value is at least $2|\mathcal{S}| + 2n + 1$. Suppose, for the sake of contradiction, that $(\mathcal{U}, \mathcal{S})$ is a NO-instance of 3-D-Matching. Consider a solution where k players in M beat exactly ℓ players in L. This implies that the maximum contribution to the tournament value from the players in L is only ℓ. Additionally, these k players can contribute at most $k - 1$ points to the tournament value by playing among themselves. An additional 2 points can be achieved if one of these players loses to h, but only one player can benefit from this. For the remaining $|\mathcal{S}| - k$ players in M, the maximum contribution can be $2(|\mathcal{S}| - k)$ points if they all lose to h.

Therefore, the total value that can be achieved is $2|\mathcal{S}| - k + 1 + \ell$. For this to be at least $2|\mathcal{S}| + 2n + 1$, we need $\ell - k \geqslant 2n$. Since we assume that the given instance is NO instance, from Proposition 4, $\ell - k < 2n$, giving a contradiction. □

[3]For a vertex subset $S \subseteq \mathcal{V}$ in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we use $G[S]$ to denote the subgraph of G induced by S.

Next, we present an approximation algorithm for player-popularity-based CTC-VM, with an approximation factor that depends on the number of distinct popularity values.

THEOREM 6. *There is a $\frac{1}{k-1}$-approximation algorithm for player-popularity-based CTC-VM where the range of the popularity values is a set of cardinality k.*

PROOF. Let the range of the popularity values be $\{v_i : i \in [k]\}$ where we have $v_1 > v_2 > \cdots > v_k$. Let P_i be the set of players whose popularity is v_i for $i \in [k]$. Without loss of generality, we can assume that $v_k = 0$. Our algorithm is as follows.

(1) For every $i \in [k-1]$, run CTC-VM, by setting the value of every player except those in P_i to be 0, using the algorithm described in Theorem 3.
(2) Among these $k-1$ tournaments, select the one with the highest value as the output.

Our algorithm runs in polynomial time since the algorithm in Theorem 3 runs in polynomial time. We next prove its approximation guarantee.

Let us consider a challenge the champ tournament T of the highest value for the input instance. Let its value be OPT. Let V_i be the sum of the values of the matches in T where a player from P_i has won for $i \in [k-1]$. Then we have

$$OPT = \sum_{i=1}^{k-1} V_i.$$

We now have the following if ALG is the value of the tournament output by the algorithm.

$$ALG \geqslant \max_{i=1}^{k-1} V_i \geqslant \frac{1}{k-1} \sum_{i=1}^{k-1} V_i = \frac{OPT}{k-1}.$$

□

4 WIN-COUNT-BASED TOURNAMENT VALUE FUNCTIONS

We now consider win-count-based value functions, modeling situations where a player's popularity changes through the course of the tournament. For these value functions, the value a player brings to a match depends on how many matches the player has won. We first prove that CTC-VM-DAG is polynomial-time solvable for win-count-based tournament value functions by giving a dynamic programming algorithm. We note that, in fact, Theorem 7 is a more general result than Theorem 2. However, it does not give an explicit value for the optimal seeding. Theorem 2 gives an explicit seeding and value, as well as a simpler greedy algorithm. Further, as discussed earlier, it allows us to show that for DAGs and player-popularity-based value functions, Challenge the Champ tournaments are optimal among all single-elimination tournaments. Due to lack of space, the proof of the following theorem is omitted and can be found in the full version of the paper [3].

THEOREM 7. *There is a polynomial-time algorithm for CTC-VM-DAG for win-count-based tournament value functions.*

Next, we show that CTC-VM is NP-hard for binary-valued win-count-based games. Recall that here, every player has a threshold λ_i, and the tournament value for a seeding σ is $|\{i : w_i(\sigma) \geqslant \lambda_i\}|$. For this, we give a polynomial-time reduction from INDEPENDENT SET. The input of INDEPENDENT SET consists of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a positive integer $k \in \mathbb{N}$. The problem is to determine if there exists a subset $\mathcal{W} \subseteq \mathcal{V}$ of size k so that every edge in $\mathcal{E}$ is incident on at most one vertex in $\mathcal{W}$.

THEOREM 8. *CTC-VM is* NP*-hard for binary-valued win-count-based tournament value functions.*

PROOF. Let $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), k)$ be a given instance of INDEPENDENT SET. Let $n = |\mathcal{V}|$, and $d(v)$ be the degree of vertex $v \in \mathcal{V}$. We construct an instance $(N, \mathcal{T}, f, t)$ of CTC-VM as

$$N = \{p_v : v \in \mathcal{V}\} \cup \{p_e^\ell : e \in \mathcal{E}, \ell \in [n^2]\}.$$

Thus there is a player p_v for every vertex in $\mathcal{G}$ whom we call a *vertex player*, and n^2 players $(p_e^\ell)_{\ell \in [n^2]}$ for every edge e in $\mathcal{E}$ whom we call *edge players*. We now describe the strength graph $\mathcal{T}$ between the players. If $v \in e$ in $\mathcal{G}$, then the player p_v beats p_e^ℓ, otherwise p_e^ℓ beats p_v. Thus a vertex player beats all edge players for incident edges, and is beaten by all other edge players. The directions of the remaining edges of the strength graph are arbitrary. An illustration of the construction of a strength graph from a given instance of INDEPENDENT SET is shown in Figure 5. The value of a vertex player p_v is 1 if she wins at least $d(v) \times n^2$ matches; otherwise, her value is zero. Thus the threshold $\lambda_{p_v} = d(v) \times n^2$. The value of all edge players is zero, irrespective of the number of matches she wins. The target value $t = k$. We claim that there is a seeding that achieves the target value if and only if the graph $\mathcal{G}$ has an independent set of size k.

In one direction, let $\mathcal{W} \subseteq \mathcal{V}$ form an independent set of size k, and assume $\mathcal{W} = \{v_1, \ldots, v_k\}$. Let $N_\mathcal{W}$ be the corresponding set of players $\{p_v : v \in \mathcal{W}\}$, and let $\mathcal{H}_\mathcal{W}$ be a Hamiltonian path on the players in $N_\mathcal{W}$. By reindexing the vertices, assume $\mathcal{H}_\mathcal{W} = (p_{v_k}, p_{v_{k-1}}, \ldots, p_1)$. Note that this implies that player $p_{v_{i+1}}$ beats player p_{v_i} in the strength graph, and by construction and since $\mathcal{W}$ is an independent set, each edge player p_e^ℓ is beaten by at most one player in $N_\mathcal{W}$. Consider the seeding that starts with player p_{v_1}, then consists of all $d(v_1) \times n^2$ edges players beaten by p_{v_1}, then has player p_{v_2} followed by all edge players beaten by p_{v_2}. We continue in this manner until p_{v_k} and all edge players are beaten by this player. The remaining players are then added in this sequence arbitrarily. Since each player p_v for $v \in \mathcal{W}$ beats $d(v) \times n^2$ edge players, it is easy to see that this seeding has the desired tournament value.

In the other direction, suppose there is a seeding with tournament value k. Clearly, there must be k vertex players that have positive value, since only vertex players can have positive value. Let $N_\mathcal{W} = \{p_{v_1}, \ldots, p_{v_k}\}$ be these vertex players, and $\mathcal{W} = \{v_1, \ldots, v_k\}$ are vertices corresponding to these players in the graph $\mathcal{G}$. We claim that the set $\mathcal{W}$ must be an independent set in $\mathcal{G}$. If not, suppose $v_i, v_j \in \mathcal{W}$ are adjacent, with edge $\hat{e} = \{v_i, v_j\} \in \mathcal{E}$. Then both p_{v_i} and p_{v_j} have outgoing edges to the n^2 edge players $p_{\hat{e}}^\ell$ in the strength graph. Consequently, at most $n + n^2(d(v_i) + d(v_j) - 1)$ players have incoming edges from either v_i or v_j,[4] and hence for large enough n, v_i and v_j cannot each beat $d(v_i) \times n^2$ and $d(v_j) \times n^2$ players respectively. It follows immediately that $\mathcal{G}$ has an independent set of size k, completing the proof. □

[4]The additional n players are due to vertex players possibly beaten by p_{v_i} and p_{v_j}.

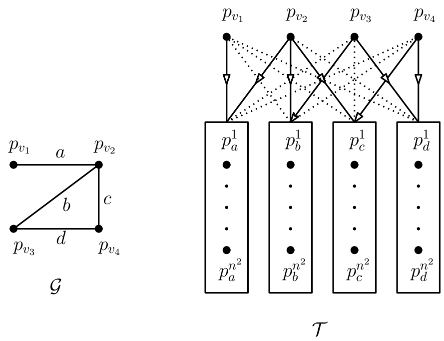

Figure 5: An edge from a vertex to a box denotes all directed edges between that vertex and vertices within the box in the same direction. Edges not depicted are arbitrarily directed. The dotted edges are directed from the vertices in the box to the outside vertex.

In the proof of Theorem 8, the maximum value of any Challenge the Champ tournament obtained in the reduction is the same as the maximum size of an independent set in the Independent Set instance. Hence, we obtain the following from the known inapproximability of Independent Set [11, 12, 27].

Corollary 9. *For every real number $\varepsilon > 0$, no polynomial-time algorithm approximates the value of CTC-VM for binary-valued win-count-based tournament value functions within a factor of $n^{1-\varepsilon}$ unless* P = NP.

We next consider linear-after-threshold value functions. Recall that now each player i has a threshold λ_i, and for a seeding σ, the value obtained is $\sum_{i \in N} \max\{0, w_i(\sigma) - \lambda_i + 1\}$. By a modification of the previous reduction, we show next that CTC-VM is NP-hard even for linear-after-threshold valuation functions. However, unlike Theorem 8, the reduction is not approximation-preserving, hence we do not obtain the same approximation hardness result. Due to lack of space, the proof of the following theorem is omitted and can be found in the full version of the paper [3].

Theorem 10. *CTC-VM is* NP*-hard for linear-after-threshold valuation functions.*

5 PAIR-BASED TOURNAMENT VALUE FUNCTIONS

For pair-based tournament value functions, we show that CTC-VM-Dag is NP-hard, even for binary valuations. Our proof gives a reduction from 3-D-Matching. The reduction is similar to the proof of Theorem 5, but the added flexibility of pair-based value functions allows us to slightly simplify the proof. Due to lack of space, the proof of the following theorem is omitted and can be found in the full version of the paper [3].

Theorem 11. *CTC-VM-Dag is NP-complete for pair-based tournament value functions when the tournament value function maps to* $\{0, 1\}$.

Consider the following problem. Given a complete directed acyclic graph $\mathcal{G}$ with binary weights $w_e \in \{0, 1\}$ on each directed edge, the problem is to find a maximum weight caterpillar arborescence — i.e., an arborescence of maximum weight such that the removal of all vertices with out-degree zero in the arborescence gives a directed path. We can show this problem is also NP-hard, since there is a direct reduction from CTC-VM-Dag with binary pair-based tournament value functions. We simply let $\mathcal{G}$ be the strength graph $\mathcal{T}$, and set the weight w_e of an edge $e = (i, j)$ to be the corresponding value for the pair of players i, j. It follows from Theorem 11 that this problem is also NP-hard.

Corollary 12. *Computing a maximum weight caterpillar arborescence in a DAG is* NP*-hard.*

6 CONCLUSION AND OPEN PROBLEMS

In this paper, we initiate the study of tournament value maximization for Challenge the Champ tournaments and contribute to the growing body of research on tournament fixing and design problems. Our results provide a complete and comprehensive picture of the computational complexity of value maximization, including for binary and ternary value functions. Theorems 2 and 3, in fact, provide upper bounds on value maximization for *all* single-elimination tournaments. En route, we show interesting connections to Hamiltonian paths in complete directed graphs and spanning caterpillar arborescences.

An obvious question for future research in value maximization is approximation. All of our hardness results are for binary or ternary valuations. These results hence show strong NP-hardness, and rule out fully polynomial time approximation schemes (FPTAS), unless P = NP. However, weaker approximations may be possible. We believe that given that value maximization is a practical concern, this is an interesting and important direction for research.

A second question is with regard to parameterized complexity. There are many relevant and natural parameters worthy of investigation. Our paper shows that for win-count-based tournament value functions, value-maximization is easy if the strength graph is a DAG but is NP-hard otherwise, even for binary values. An obvious parameter to try is then the size of the *feedback vertex set* or the *feedback arc set* of the strength graph. Also, for player-popularity-based games with ternary values, the problem is NP-hard even with a single player with value 2. It is possible that using the number of 0 or 1 value players as a parameter would be helpful.

Lastly, value maximization in other tournament formats, including knockout tournaments and extended stepladder tournaments, has only received limited attention and presents a rich and important avenue for further research.

ACKNOWLEDGMENTS

Umang Bhaskar and Juhi Chaudhary would like to thank the Department of Atomic Energy, Government of India, under project no. RTI4001. Palash Dey thanks the Science, Education, and Research Board (SERB), Government of India, for supporting this work through Core Research Grant under file no. CRG/2022/003294.

REFERENCES

[1] Ritxar Arlegi and Dinko Dimitrov. 2020. Fair elimination-type competitions. *Eur. J. Oper. Res.* 287, 2 (2020), 528–535.

[2] Haris Aziz, Serge Gaspers, Simon Mackenzie, Nicholas Mattei, Paul Stursberg, and Toby Walsh. 2018. Fixing balanced knockout and double elimination tournaments. *Artificial Intelligence* 262 (2018), 1–14.

[3] Umang Bhaskar, Juhi Chaudhary, and Palash Dey. 2025. Maximizing Value in Challenge the Champ Tournaments. *arXiv preprint arXiv:2502.12569* (2025).

[4] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. 2016. *Knockout Tournaments*. Cambridge University Press.

[5] Juhi Chaudhary, Hendrik Molter, and Meirav Zehavi. 2024. How to Make Knockout Tournaments More Popular?. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9582–9589.

[6] Juhi Chaudhary, Hendrik Molter, and Meirav Zehavi. 2024. Parameterized analysis of bribery in challenge the champ tournaments. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2704–2712.

[7] Juhi Chaudhary, Hendrik Molter, and Meirav Zehavi. 2025. Adaptive Manipulation for Coalitions in Knockout Tournaments. *arXiv preprint arXiv:2412.11799v1* (2025).

[8] Dmitry Dagaev and Alex Suzdaltsev. 2018. Competitive intensity and quality maximizing seedings in knock-out tournaments. *Journal of Combinatorial Optimization* 35 (2018), 170–188.

[9] Michael R Garey and David S Johnson. 1979. *Computers and intractability*. Vol. 174. freeman San Francisco.

[10] Sushmita Gupta, Ramanujan Sridharan, and Peter Strulo. 2024. An Exercise in Tournament Design: When Some Matches Must Be Scheduled. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9749–9756.

[11] Johan Håstad. 1996. Clique is Hard to Approximate Within $n^{1-epsilon}$. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*. IEEE Computer Society, 627–636. https://doi.org/10.1109/SFCS.1996.548522

[12] Subhash Khot. 2001. Improved Inaproximability Results for MaxClique, Chromatic Number and Approximate Graph Coloring. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. IEEE Computer Society, 600–609. https://doi.org/10.1109/SFCS.2001.959936

[13] Nicholas Mattei, Judy Goldsmith, Andrew Klapper, and Martin Mundhenk. 2015. On the complexity of bribery and manipulation in tournaments with uncertain information. *Journal of Applied Logic* 13, 4 (2015), 557–581.

[14] Roland Pongou, Bertrand Tchantcho, and Narcisse Tedjeugang. 2017. Properties of ladder tournaments. *European Journal of Operational Research* 263, 1 (2017), 203–213.

[15] László Rédei. 1934. Ein kombinatorischer satz. *Acta Litt. Szeged* 7, 39-43 (1934), 97.

[16] KB Reid. 1991. Majority tournaments: sincere and sophisticated voting decisions under amendment procedure. *Mathematical Social Sciences* 21, 1 (1991), 1–19.

[17] K Brooks Reid and Nicholas Charles Wormald. 1983. Embedding oriented n-trees in tournaments. *Studia Sci. Math. Hungar* 18, 2-4 (1983), 377–387.

[18] Tyrel Russell and Toby Walsh. 2009. Manipulating tournaments in cup and round robin competitions. In *Algorithmic Decision Theory: First International Conference, ADT 2009, Venice, Italy, October 20-23, 2009. Proceedings 1*. Springer, 26–37.

[19] Pascal Sauer, Ágnes Cseh, and Pascal Lenzner. 2024. Improving ranking quality and fairness in Swiss-system chess tournaments. *Journal of Quantitative Analysis in Sports* 20, 2 (2024), 127–146.

[20] Isabelle Stanton and Virginia Vassilevska Williams. 2013. The structure, efficacy, and manipulation of double-elimination tournaments. *Journal of Quantitative Analysis in Sports* 9, 4 (2013), 319–335.

[21] Warut Suksompong. 2016. Scheduling asynchronous round-robin tournaments. *Operations Research Letters* 44, 1 (2016), 96–100.

[22] Warut Suksompong. 2021. Tournaments in Computational Social Choice: Recent Developments. In *IJCAI*. 4611–4618.

[23] Balázs R Sziklai, Péter Biró, and László Csató. 2022. The efficacy of tournament designs. *Computers & Operations Research* 144 (2022), 105821.

[24] Wikipedia contributors. 2024. Tournament — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Tournament&oldid=1239081863 [Online; accessed 16-October-2024].

[25] Virginia Vassilevska Williams. 2016. *Knockout Tournaments*. Cambridge University Press, 453–474. https://doi.org/10.1017/CBO9781107446984.020

[26] Yongjie Yang and Dinko Dimitrov. 2021. Weak transitivity and agenda control for extended stepladder tournaments. *Economic Theory Bulletin* 9 (2021), 27–37.

[27] David Zuckerman. 2006. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, Jon M. Kleinberg (Ed.). ACM, 681–690. https://doi.org/10.1145/1132516.1132612