The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

Adaptive Manipulation for Coalitions in Knockout Tournaments

Juhi Chaudhary[1], Hendrik Molter[2], Meirav Zehavi[2]

[1]School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India.
[2]Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel.Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel.
juhi.chaudhary@tifr.res.in, molterh@post.bgu.ac.il, meiravze@bgu.ac.il

Abstract

Knockout tournaments, also known as *single-elimination* or *cup tournaments*, are a popular form of sports competitions. In the standard probabilistic setting, for each pairing of players, one of the players wins the game with a certain (a priori known) probability. Due to their competitive nature, tournaments are prone to manipulation. We investigate the computational problem of determining whether, for a given tournament, a coalition has a manipulation strategy that increases the winning probability of a designated player above a given threshold. More precisely, in every round of the tournament, coalition players can strategically decide which games to throw based on the advancement of other players to the current round. We call this setting *adaptive constructive coalition manipulation*. To the best of our knowledge, while coalition manipulation has been studied in the literature, this is the first work to introduce adaptiveness to this context.

We show that the above problem is hard for every complexity class in the polynomial hierarchy. On the algorithmic side, we show that the problem is solvable in polynomial time when the coalition size is a constant. Furthermore, we show that the problem is fixed-parameter tractable when parameterized by the coalition size and the size of a minimum player set that must include at least one player from each non-deterministic game. Lastly, we investigate a generalized setting where the tournament tree can be imbalanced.

1 Introduction

Knockout tournaments, also known as *single-elimination* or *cup tournaments*, constitute a competition format in which contestants are paired up and compete against each other in rounds, with the losers being eliminated after each round. The tournament continues until only one contestant remains, the *winner*. This format is widely favored in sports (Chaudhary, Molter, and Zehavi 2024b; Connolly and Rendleman 2011; Groh et al. 2012; Suksompong 2021; Williams 2016) and finds applications in several other areas, such as elections and decision-making processes (Vu, Altman, and Shoham 2009; Laslier 1997; Brandt and Fischer 2007; Tullock 1980; Rosen 1985). More formally, a knockout tournament consists of n players, where we assume that n is a power of two, and a bijective mapping, called a *seeding*, of the players to the leaves of a complete binary tree. As long as at least two players remain, every two players mapped to leaves having the same parent in the tree play a match, whose winner is mapped to the common parent; then, the leaves of the tree are deleted, which means that the losers are knocked out of the tournament. When exactly one player remains, it is declared the winner.

Due to their competitive nature, knockout tournaments are highly prone to various forms of manipulation. Such manipulations are frequently reported by the media (see, e.g., (MediaNews c,b,a; Manoli and Antonopoulos 2015; Hill 2010; Feltes 2013)), and their vulnerability was shown empirically (see, e.g., (Stanton and Williams 2013; Mattei and Walsh 2016)). The focus in almost all research on manipulation in knockout tournaments in the literature is on *constructive manipulation*, where we aim to make a favorite player win the competition. Here, the most well-studied forms of manipulation include coalition manipulation (discussed below), and tournament fixing where, generally, the manipulation involves selecting the seeding (Gupta et al. 2018a, 2019; Kim, Suksompong, and Williams 2017; Aziz et al. 2018; Zehavi 2023; Gupta et al. 2018b; Vu, Altman, and Shoham 2009; Williams 2010), and bribery where we can flip the outcome of a bounded, or budget-constrained, number of games (Konicki and Williams 2019; Mattei et al. 2015; Kim and Williams 2015; Saarinen, Goldsmith, and Tovey 2015).

We focus on constructive *coalition manipulation* in knockout tournaments. Constructive coalition manipulation is well-studied in various forms of tournaments (including knockout tournaments) as well as in various settings in voting theory, as discussed in Section 1. Here, generally, a given subset of players is termed a *coalition*. Then, the objective is to determine whether the coalition has a "strategy"—where coalition players intentionally lose some of their games—so that the favorite player will win or will be likely to win. The simplest setting for such manipulation, considered by Russell and Walsh (Russell and Walsh 2009), is when the outcome of every potential match is deterministic and known in advance, i.e., for every two players, we know for certain who will win a match between them, supposing none manipulates, and when we define a strategy as the set of matches that the coalition players should lose (when a match involves two coalition players, we further specify who among them should lose). Russell and Walsh (Russell and Walsh 2009) considered the computational problem of determining, in this setting, whether there exists a strategy for the coalition players that makes the favorite player win and showed that it is solvable in polynomial time.

Mattei et al. (Mattei et al. 2015) study coalition manipulation in the more general case where games have probabilistic outcomes. In their model, the coalition players may modify their winning probabilities before the start of the tournament so that the favorite player will have a higher chance of winning the tournament. However, we will make an argument that it is never non-optimal for a coalition player to either stay at their original winning probability or change it to zero. Hence, the two models may be considered equivalent. Mattei et al. (Mattei et al. 2015) show that the computational problem of determining whether there exists a strategy for the coalition players such that the favorite player wins the tournament with at least some given probability is contained in NP. They leave open whether this problem is NP-hard.

Our Model. While the simple setting of Russell and Walsh (Russell and Walsh 2009) discussed above is an important entry point, it is unrealistic in one way and too restrictive in another. Specifically, on the one hand, it is rarely conceivable that the outcomes of all possible matches will be deterministic—i.e., that we can always be *completely certain*, in advance, who beats whom. To overcome this, we employ the standard *probabilistic model* for knockout tournaments, which is also used by Mattei et al. (Mattei et al. 2015). Here, for every two players, we know some estimate of the probability of one beating the other. Such estimations can often be derived from available statistics regarding past matches; see, e.g., (ESPN; Predict; Team; 538).

On the other hand, it is completely unnecessary to make all decisions regarding which manipulations to perform before the tournament has even started. Of course, if the tournament is entirely deterministic, then the timing of these decisions is immaterial. However, when probabilities are involved, the situation changes drastically. When we reach a certain round of the tournament, the coalition players observe what is the *current* seeding, that is, which players advanced to this round. According to this crucial information, the coalition players decide which matches to lose in the current round. That is, decisions are made on a round-by-round basis, making use of as much available information as possible. In other words, the strategy is *adaptive*. To the best of our knowledge, while coalition manipulation has been studied in the literature, this is the first work to introduce adaptiveness to this context.

Thus, we introduce a new computational problem termed Adaptive Constructive Coalition Manipulation for Knockout Tournaments (ACCM-KT). Here, the input consists of: N, a set of n players, a seeding of N, for every two players, the probability of one beating the other, a coalition $C \subseteq N$, a favorite player $e^* \in N$, and a probability $t \in [0, 1]$. Then, the objective is to determine whether there exists an *adaptive strategy* (formally defined in Section 2) for the coalition players so that the probability that e^* will be the winner is at least t. We assume that the non-coalition players do not behave strategically, that is, the outcome probability of all games not involving coalition players is given in the input.

Of course, the answer to the question above does not yield the strategy itself. Moreover, an explicit description of the strategy is of exponential size, since it encodes, for every round and every *possible* advancement of players to that round (of which there exist exponentially many), what should the coalition players do? Fortunately, there is no utility in knowing the entire strategy, as almost all of it concerns hypothetical situations that will not occur. Practically, what the coalition players need to know is what to do in a current round, to which, in particular, we know exactly who are the players that advanced. Thus, we also consider the accompanying *best response* version of the problem, where the objective is to compute only the part of the strategy that concerns the tournament's first round. Once the second round is about to start—and then we know who advanced to it—we can simply recompute the best response for the remaining tournament, treated as the input tournament, and so on for the later rounds. We term this problem Best Response for ACCM-KT (BR-ACCM-KT).

In addition to ACCM-KT and BR-ACCM-KT, we also investigate the generalized versions of these problems where the tournament tree can be imbalanced. That is, the tournament tree is still binary, but the distance between the root to leaves can differ. A formal definition can be found in Section 2. We refer to the generalized problems as Adaptive Constructive Coalition Manipulation for Generalized Knockout Tournaments (ACCM-GKT) and Best Response for ACCM-GKT (BR-ACCM-GKT). Lastly, one of our results also resolves an open case for the non-adaptive version of ACCM-KT, which we call Constructive Coalition Manipulation for Knockout Tournaments (CCM-KT).

Our Contribution. In addition to the introduction of our adaptive model and corresponding computational problems in Section 3, we prove several highly non-trivial technical contributions, described below. We present the hardness results in Section 4 and the algorithmic results in Section 5. We use the standard concepts and notations from classical and parameterized complexity theory (Arora and Barak 2009; Downey, Fellows et al. 2013; Cygan et al. 2015), more information is deferred to the full version (Chaudhary, Molter, and Zehavi 2024a). Due to space constraints, some of the proof details are deferred to the full version (Chaudhary, Molter, and Zehavi 2024a). Some of the questions left open are discussed in Section 6.

We first establish the classical hardness of our problems:

- ACCM-KT is hard for each class in the polynomial hierarchy (PH).
- ACCM-GKT is PSPACE-hard.
- BR-ACCM-KT is NP-hard.
- CCM-KT is NP-hard. (This case was left open by Mattei et al. (Mattei et al. 2015).)

Due to their computational hardness, we develop parameterized algorithms for our problems. As parameters, we consider the coalition size $|C|$ and the size x of a minimum *random game cover*, that is, the size of a minimum player set that must include at least one player from each non-deterministic game (a formal definition is given in Section 3). Specifically, we prove that:

- ACCM-GKT can be solved in $n^{O(|C|)}$ time.
- ACCM-KT can be solved in $(|C| + x)^{O(|C|+x)} \cdot n^{O(1)}$ time.
- ACCM-GKT is contained in PSPACE.

Furthermore, we show that the algorithms can also be used to compute best responses. Formally, we show the following.

- BR-ACCM-GKT can be solved in $n^{O(|C|)}$ time and BR-ACCM-GKT can be solved in polynomial space.
- BR-ACCM-KT can be solved in $(|C| + x)^{O(|C|+x)} \cdot n^{O(1)}$ time.

Lastly, we complement the aforementioned results by proving the following parameterized hardness result.

- ACCM-GKT is NP-hard and W[1]-hard when parameterized by the coalition size even if $x = 2$.

Additional Related Works. Russell and Van Beck (2012) investigated the problem of developing automated tools for detecting coalitions of teams manipulating the winner in both knockout and round-robin competitions. In the realm of double-elimination tournaments (DETs), Stanton and Williams (2013) established that coalition manipulation of DETs can be polynomially computed under certain restrictions. Here, coalitions of players are vulnerable to manipulation by throwing matches, a phenomenon recently observed in Olympic Badminton. Schneider *et al.* (2016) also discussed manipulation by coalitions through a collusion between several teams. Shifting the focus to voting, Walsh (2011) explored the computational complexity of assessing manipulation potential in weighted voting systems, where agents may manipulate outcomes by misrepresenting their preferences. Durand (2015) posed the question of whether a strategically voting subset could elect a preferred candidate over the truthful voting outcome. Even simulations on empirical data have been done by Durand (2023). Yang (2023) investigated coalition manipulation, revealing that the two-stage majoritarian rule is resilient against most control challenges but susceptible to coalition manipulation.

A brief discussion of other works on various forms of manipulation in knockout tournaments and beyond (see (Chaudhary, Molter, and Zehavi 2024c; Saarinen, Goldsmith, and Tovey 2015)) is deferred to the full version (Chaudhary, Molter, and Zehavi 2024a).

2 Preliminaries on Knockout Tournaments

We begin by defining the following model. A tournament has a set of n *players* $N = \{e_1, \ldots, e_n\}$, where for simplicity, we assume that n is a power of two, that is, $n = 2^r$. Let $\mathbb{Q}^{n\times n}_{[0,1]}$ be the set of $n \times n$ matrices over $[0, 1] \cap \mathbb{Q}$ (the rational numbers in the interval $[0, 1]$). Let $p(i, j)$ denote the probability that player e_i will defeat player e_j. We denote with $P_N = [p(i, j)]_{i,j\in[n]\times[n]} \in \mathbb{Q}^{n\times n}_{[0,1]}$ the *probability matrix* for the tournament. Note that $p(i, j) + p(j, i) = 1$. We call a permutation vector $s_1 \in N^n$, that is, a vector of length n that contains every player in N exactly once, a *tournament seeding*. Let $r = \log n$ be the number of rounds of the tournament. Then we call a permutation vector $s_k \in N^{2^{(r-k+1)}} \times \{\perp\}^{2^r-2^{r-k+1}}$ a *seeding for the kth round* of the tournament (which has $2^{(r-k+1)}$ players). Note that a seeding for round k is a vector of the same length as the tournament seeding. The $\perp$-symbol indicates that the seed positions are not present. A tournament seeding determines how to label the n leaves of an ordered complete binary tree with the players.

Given a set of players N and a seeding s_1 of the players, we define a *tournament tree* T to be an ordered rooted complete binary tree with n leaves, where leaves are labeled with the players according to the seeding s_1. More formally, the ith leaf of the tournament tree is labeled with player e_j if and only if $s_1(i) = j$. For a vertex v in T that is not the root of the tree, we call v' the *sibling* of v if v and v' have the same parent vertex in T. Given a seeding, the competition is conducted in rounds as follows. As long as the tournament 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. Let s_k be a seeding for the kth round of the tournament. If in round k of the tournament, the leaves of the binary tree are labeled according to s_k, then we say that s_k is obtained in round k.

3 Our Model: Adaptive Manipulation

A *coalition* $C \subseteq N$ is a subset of players. A coalition player can intentionally lose a game, that is, set their winning probability to zero. A *strategy* for the coalition players is a function $\xi : C \times [\log n] \times (N \cup \{\perp\})^n \to \{0, 1\}$ that takes a coalition player, a round number, and a seeding for that round as input, and outputs zero or one. Let e be a coalition player, let k be a round number, and let s_k be a seeding for round k of the tournament. If $\xi(e, k, s_k) = 0$, then player e intentionally loses their game in round k of the tournament if the seeding s_k is obtained in round k. Otherwise, player e tries to win their game. If two coalition players play against each other, then not both of them can intentionally lose the game. Formally, we consider the following problem.

> ADAPTIVE CONSTRUCTIVE COALITION MANIPULATION FOR KNOCKOUT TOURNAMENTS (ACCM-KT)
>
> Input: A set of players N, a coalition $C \subseteq N$, a favorite player $e^* \in N$, a probability matrix P_N, a tournament seeding s_1, and a probability threshold $t \in [0, 1]$.
>
> Question: Is there a strategy ξ for the coalition players such that when applied the winning probability of e^* for the whole tournament is at least t?

We remark that Mattei et al. (Mattei et al. 2015) allow coalition players in their model to lower their probabilities arbitrarily. However, we argue in the full version (Chaudhary, Molter, and Zehavi 2024a) that there are always optimal strategies where coalition players either stay at their original winning probability or lower it to zero.

Best Response. In application settings, we might want to output an actual strategy that the coalition players can use rather than only solving the decision problem. To this end, we introduce the so-called *best response* problem. Intuitively, given a certain round seeding s_k for a tournament round k, the best response tells the coalition players which strategy to use in this round to maximize the winning probability of e^*. After the first k rounds of the tournament are played out, we can then compute the best response for the next round. This should give e^* the optimal chance to win the tournament. To define the best responses formally, we need to know what the best possible winning probability for e^* is in a given tournament. Let N be a set of players (with $e^* \in N$), let $C \subseteq N$ be a coalition, let P_N be a probability matrix, and let s_1 be a tournament seeding. We define the *best possible winning probability*

$$t_{\text{opt}} = \arg\max_{t \in [0,1]} \{(N, C, e^*, P_N, s_1, t) \text{ is a yes-instance}\}.$$

The best possible winning probability $t^k_{\text{opt}}(s_k)$ after round k and a given seeding s_k for the kth round is the best possible winning probability of the remaining tournament. Formally, we remove all players that are knocked out from the tournament, and we remove all $\perp$ entries from the seeding for the kth round to obtain a new (smaller) instance of ACCM-KT.

Next, we define a *strategy profile*. It is a function $c : C \to \{0, 1\}$. If $c(e) = 0$ for some $e \in C$, we interpret this as e manipulating (that is, intentionally losing) when strategy profile c is used in a certain round of the tournament. A *best response* for a tournament with players N, coalition C, probability matrix P_N, and tournament seeding s_1 is a strategy profile c_{best} with the following property. Let $p(c, s_1, s_2)$ denote the probability that s_2 is obtained as a seeding for the second round of the tournament when strategy profile c and tournament seeding s_1 is used in the first round. Then the strategy profile c_{best} is a best response if we have

$$\sum_{s_2} p(c_{\text{best}}, s_1, s_2) \cdot t^{(2)}_{\text{opt}}(s_2) = t_{\text{opt}}.$$

Now, we can define the following computational problem.

BEST RESPONSE FOR ACCM-KT (BR-ACCM-KT)	
Input:	A set of players N, a coalition $C \subseteq N$, a favorite player $e^* \in N$, a probability matrix P_N, and a tournament seeding s_1.
Task:	Compute a best response c_{best} for the first round of the tournament.

Non-Adaptive Problem. In the non-adaptive problem, all coalition players have to specify which games they intend to lose before the tournament starts, called CONSTRUCTIVE COALITION MANIPULATION FOR KNOCKOUT TOURNAMENTS (CCM-KT). The formal definition is deferred to the full version (Chaudhary, Molter, and Zehavi 2024a).

Generalized Problem. Furthermore, we investigate a generalized version of the problem where we do not require the tournament to be balanced, called ADAPTIVE CONSTRUCTIVE COALITION MANIPULATION FOR GENERALIZED KNOCKOUT TOURNAMENTS (ACCM-GKT). More formally, we allow as a tournament tree every ordered rooted binary tree, that is, an ordered rooted tree where every vertex either has two descendants or is a leaf. We can, in an analogous way, also define a best response for the generalized setting. This leads to the problem BEST RESPONSE FOR ACCM-GKT (BR-ACCM-GKT). Due to space constraints, the formal definitions of these problems are deferred to the full version (Chaudhary, Molter, and Zehavi 2024a).

Random Game Covers. Intuitively, a *random game cover* is a set of players such that for every game with a non-deterministic outcome, at least one player of the random game cover is involved. Formally, a (minimum) random game cover is defined as follows.

Definition 1. A *random game cover* for a tournament is a set $X \subseteq N$ of players such that for all pairs of players i, j such that $0 \neq p(i, j) \neq 1$ we have that $\{i, j\} \cap X \neq \emptyset$, that is, $i \in X$ or $j \in Y$. A *minimum random game cover* is a random game cover of minimal cardinality.

We can observe that computing a minimum random game cover is equivalent to computing a minimum vertex cover of the graph with the set of players as vertices and where two players i, j are adjacent if and only if $0 \neq p(i, j) \neq 1$. Hence, we get the following (Cygan et al. 2015).

Proposition 2. *A minimum random game cover can be computed in $2^{O(x)} \cdot n^{O(1)}$ time, where x is the size of a minimum random game cover.*

4 Hardness Results

Classical Hardness Results. The following results are obtained by reductions from QUANTIFIED BOOLEAN FORMULA (Arora and Barak 2009) with different constraints on the quantifiers and the number of alternations. Due to space constraints, we do not present technical details here and defer the full proofs to the full version (Chaudhary, Molter, and Zehavi 2024a).

Theorem 3. *ACCM-KT is hard for each class in the PH.*

Theorem 4. *ACCM-GKT is PSPACE-hard.*

Theorem 5. BR-ACCM-KT *and CCM-KT are NP-hard.*

The main ingredient for the reductions used to prove Theorems 3 to 5 is a small (constant-sized) subtournament, where a coalition player can decide which of the other players wins the subtournament. We use these subtournaments to create *variable gadgets* for the existentially quantified variables and *clause gadgets*. Intuitively, in the former, the coalition player can select an assignment for an existentially quantified variable, and in the latter, the coalition player can select a literal of the clause (we can assume that each clause has three literals). In the simplest case, where there is only one existential quantifier, we arrange the gadgets (and define the winning probabilities) in a way that each winner of a variable gadget has a non-zero probability of reaching the semi-final game, and each winner of a clause gadget has a non-zero probability of reaching the same semi-final game.

Then, the clause player beats the variable player with probability one if the clause player corresponds to a literal that involves the variable corresponding to the variable player and the variable player represents a truth assignment to the variable that does *not* satisfy the literal. Otherwise, the variable player beats the clause player with probability one. In the final game, the winner of the variable and the clause player meets e^*, who loses against the clause player with probability one and wins against the variable player with probability one. This way, intuitively, the overall winning probability of e^* is one only if the coalition can prevent a clause player from reaching the finals. This only happens if a satisfying assignment for the formula is "selected" in the variable gadgets and satisfied literals are "selected" in the clause gadgets.

To obtain hardness for a class in the PH or PSPACE, informally, we have variable gadgets at different rounds of the overall tournament, alternatingly with players corresponding to universally quantified variables, who randomly advance (each with a non-zero probability) such that every possible assignment has a non-zero probability to occur. This allows coalition players in later variable gadgets to react to which players corresponding to universally quantified variables have advanced. This allows coalition players in later variable gadgets to react to the players corresponding to universally quantified variables that have advanced. As a consequence, the depth of the tournament depends linearly on the number of quantifier alternations. It follows that if the tournament needs to be balanced, it becomes exponentially large in the number of quantifier alternations. Intuitively, this is the reason why we can "only" show hardness for the PH classes for ACCM-KT while we can show PSPACE-hardness for ACCM-GKT.

Parameterized Hardness. The following parameterized hardness result is obtained by a parameterized reduction from MULTICOLORED CLIQUE (Fellows et al. 2009). Due to space constraints, we do not present technical details here and defer the full proofs to the full version (Chaudhary, Molter, and Zehavi 2024a).

Theorem 6. *ACCM-GKT is NP-hard and W[1]-hard when parameterized by the coalition size even if the size of a minimum random game cover is two.*

Similarly to the reductions for the classical hardness results, the main ingredient used to prove Theorem 6 is a selection gadgetwhich is a subtournament with one coalition player who can decide who of the other players wins. The main difference is that the gadget is not a balanced tournament, therefore it can have a polynomial number of players. Intuitively, this is also the reason why our hardness result only holds for ACCM-GKT and not ACCM-KT. We use one of these gadgets to select a vertex of each color and one of these gadgets to select an edge for each color combination. Note that this way, the number of coalition players is upper-bounded by the number of colors of the MULTICOLORED CLIQUE instance. The gadgets are now arranged in a way that each winner of a variable gadget has a non-zero probability of reaching the semi-final game, and each winner of a clause gadget has a non-zero probability of reaching the same semi-final game. To do this, we use two *randomize gadgets* (one for the selected vertices and one for the selected edges) that each contains one player that is involved in all games with non-deterministic outcomes. This way, we obtain a random game cover of size two. In the semi-final game, the vertex player beats the edge player with probability one if the edge is from a color combination that involves the color of the vertex and the vertex is *not* an endpoint of the edge. Otherwise, the vertex player beats the edge player with probability one. In the final game, the winner of the vertex and the edge player meets e^*, who loses against the edge player with probability one and wins against the vertex player with probability one. This way, intuitively, the overall winning probability of e^* is one only if the coalition can prevent an edge player from reaching the final game. This can only happen if vertices that form a clique are selected in the vertex selection gadgets and the edges in that clique are selected in the edge selection gadgets.

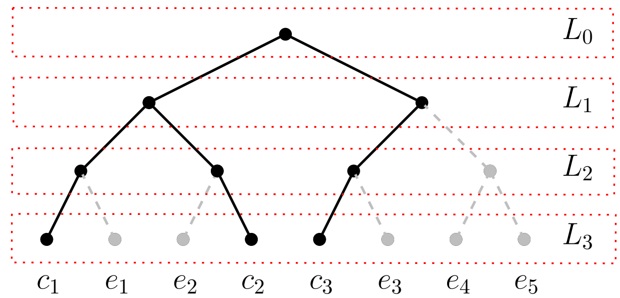

Figure 1: Here, the black edges are the edges of the coalition skeleton T_C of T, when players c_1, c_2, and c_3 are identified as coalition players. Note that for L_1, (c_1, c_3) and (c_2, e_4) are examples of valid configurations while (c_3, e_4) is not a valid configuration. Furthermore, for L_2, both (e_4) and (e_5) represent valid sibling configurations.

5 Algorithmic Results

In this section, we present our algorithmic results. The main contribution is a dynamic programming algorithm that we analyze in several different ways. From this, we obtain that ACCM-GKT is in XP when parameterized by the coalition size and that ACCM-KT is fixed-parameter tractable when parameterized by the combination of the coalition size and the size of a minimum random game cover. Furthermore, we get that ACCM-GKT is contained in PSPACE. Finally, we argue that the presented algorithm can also be used to compute the best responses.

The Dynamic Program. We start by describing a dynamic programming algorithm. To this end, we need to introduce some concepts first. Given a set of players $\{e_1, e_2, \ldots, e_n\}$ and a coalition $C \subseteq N$, we define the *coalition skeleton* T_C to be the subtree of the tournament tree T consisting of all paths from the root of T to a leaf that is labeled with a player in C. We call the ith *level* L_i of the coalition skeleton T_C the set of vertices in T_C that have distance i from the root of T_C. See Fig. 1 for an illustration. Note that every level of T_C contains at most $|C|$ vertices.

We call a subset S of players a *valid configuration* of the ith level L_i of T_C if $|S| = |L_i|$ and for every vertex $v \in L_i$, there is a player $e \in S$ such that the subtree of T rooted at v has a leaf that is labeled with e. We say that v is the position of e in L_i. Intuitively this means, that if player e reaches the $(r-i)$th round of the tournament, then e is mapped to v in the seeding for round $r-i$. We call a subset S of players a valid *sibling configuration* of the ith level L_i of T_C if $|S|$ equals the number of vertices in L_i that have a sibling which is not contained in L_i and if for every vertex $v \in L_i$ that has a sibling $v' \notin L_i$ there is a player in $e \in S$ such that the subtree of T rooted at v' has a leaf that is labelled with e. We say that v' is the position of e in L_i. It is crucial to note that valid sibling configurations do not include players from the coalition set C. Let S be a valid configuration for L_i and let S' be a valid sibling configuration for L_i. For players $e, e' \in S \cup S'$, we say that players e and e' are siblings if v is the position of e in L_i and v' is the position of e' in L_i, and v and v' are siblings in T. If players e and e' are siblings we denote $e' = s_{S,S'}(e)$. The probability $p(S)$ of a sibling configuration S for level L_i is the probability that all players in S advance to the $(r-i)$th round of the tournament, where $r \leq n$ is the total number of rounds of the tournament. Note that $p(S)$ can be computed by a straightforward dynamic program.

Given a valid configuration S for L_i, a valid sibling configuration S' for L_i, a valid configuration $S^\star$ for L_{i-1}, and a strategy profile c, we denote with $p(S, S', S^\star, c)$ the probability that configuration $S^\star$ is obtained for L_{i-1} assuming configuration S and sibling configuration S' are present in L_i and strategy profile c is used by the coalition players in round $r-i$ of the tournament. Formally, we have

$$p(S, S', S^\star, c) = \prod_{e \in S^\star \cap C} c(e) \cdot p(e, s_{S,S'}(e)) \cdot \prod_{e \in S^\star \setminus C} p(e, s_{S,S'}(e)).$$

Let $\mathcal{S} \subseteq 2^{\{e_1, e_2, \ldots, e_n\}}$ be the set of all possible configurations. We create a dynamic program $M : \{0, 1, 2, \ldots, r\} \times \mathcal{S} \to [0, 1]$ that maps a level L_i of T_C together with a valid configuration S for L_i to the probability that $e^\star$ wins assuming the players in S are seeded directly into their positions in L_i and assuming the coalition players maximize the winning probability of $e^\star$. We define M as follows. We have

$$M[0, \{e^\star\}] = 1,$$

and for all $S \subseteq \{e_1, e_2, \ldots, e_n\}$ with $S \neq \{e^\star\}$, we have

$$M[0, S] = 0.$$

Let $S \subseteq \{e_1, e_2, \ldots, e_n\}$ be valid for L_i for $i > 0$, then we have Equation (1) in Fig. 2.

Before we analyze the running time of the algorithm, we prove that the presented dynamic programming algorithm is correct.

Lemma 7. *For all $i \in \{0, 1, 2, \ldots, r\}$ and all valid configurations $S \subseteq \{e_1, e_2, \ldots, e_n\}$ for L_i we have that $M[i, S]$ is the probability that $e^\star$ wins assuming the players in S are seeded directly into their positions in L_i.*

Proof. We prove the lemma statement by induction on the level i. For $i = 0$, we have $M[0, \{e^\star\}] = 1$, and for all $S \subseteq \{e_1, e_2, \ldots, e_n\}$ with $S \neq \{e^\star\}$, we have $M[0, S] = 0$. Hence, for $i = 0$, the dynamic program is clearly correct.

Assume $i > 0$. Recall that the configuration S specifies the player positions within L_i. In order to compute a configuration for the next round of the tournament, that is, level $i-1$, we need to know which players the players in S play against. This is specified by the sibling configuration S'. The probability of attaining sibling configuration S' is represented by $p(S')$. Given a configuration S and a sibling configuration S' for level i, the probability of player advancement relies on the strategic choices made by the coalition players. Since the coalition players may choose their strategy, we consider the strategy profile that maximizes e^*'s winning probability.

Given a configuration S, a sibling configuration S', and a strategy profile c, the probability of achieving configuration $S^\star$ in level $i-1$ is denoted as $p(S, S', S^\star, c)$. By induction hypothesis, the probability that $e^\star$ wins given that $S^\star$ is the configuration of L_{i-1} is $M[i-1, S^\star]$.

The above arguments can directly be translated into the recursive formula for $M[i, S]$. Consequently, we can conclude that the lemma statement is true. □

Running Time Analysis Depending on the way we analyze the running time of the dynamic programming algorithm, we obtain several results. We first show that ACCM-GKT is contained in XP when parameterized by the coalition size. Theorem 6 implies that we cannot expect to obtain an FPT-algorithm for this parameterization.

Theorem 8. *ACCM-GKT can be solved in $n^{O(|C|)}$ time.*

Proof. Observe that there is exactly one valid configuration for level r, namely, C. By Lemma 7, it follows that $M[r, C]$ is the probability that $e^\star$ wins the tournament. In the remainder, we show that the time required to compute all entries of M (and hence, in particular, the entry $M[r, C]$) is in $n^{O(|C|)}$.

First, note that the size of the dynamic programming table is in $O(n^{|C|} \cdot n)$, since all valid configurations contain at most $|C|$ players and there are at most n rounds in the tournament. It remains to analyze the time needed to compute one entry of the table. The first sum in the recursive expression sums over all valid sibling configurations. A sibling configuration is a subset of the players of size at most $|C|$. There are $n^{O(|C|)}$ such subsets, and we can check for each one in polynomial time whether it is a valid sibling configuration. Next, we have a maximum over all strategy profiles, of which there are $2^{|C|}$ many. Lastly, we sum over all valid configurations for the previous level. There are $n^{O(|C|)}$ such configurations. Furthermore, the functions p (note that we overloaded notation here) can clearly be computed in polynomial time. It follows that the overall time required to compute all entries of M is in $n^{O(|C|)}$. □

Next, we show how to obtain an FPT-algorithm for ACCM-KT when parameterized by the combination of the coalition size and the size of a minimum random game

$$M[i,S] = \sum_{\substack{\text{valid sibling configuration} \\ S' \text{ for } L_i}} \left(p(S') \cdot \max_{\substack{\text{strategy profile} \\ c}} \left(\sum_{\substack{\text{valid configuration} \\ S^\star \text{ for } L_{i-1}}} p(S,S',S^\star,c) \cdot M[i-1,S^\star] \right) \right). \quad (1)$$

Figure 2: Equation (1) of the dynamic program.

cover. Theorem 6 implies that we cannot expect to get an FPT-algorithm with the same parameterization.

Theorem 9. *ACCM-KT can be solved in* $(|C|+x)^{O(|C|+x)} \cdot n^{O(1)}$ *time, where* x *is the size of a minimum random game cover.*

Proof. Again, as in the proof of Theorem 8, observe that there is exactly one valid configuration for level r, namely, C. Furthermore, in the case of ACCM-KT, we have that $r = \log n$. By Lemma 7, it follows that $M[\log n, C]$ is the probability that $e^\star$ wins the tournament. In the remainder, we show that the time required to compute $M[\log n, C]$ is in $(|C|+x)^{O(|C|+x)} \cdot n^{O(1)}$.

To this end, we present an alternative way to enumerate all *reachable* (sibling) configurations. More precisely, a valid sibling configuration S' is reachable if $p(S') \neq 0$. Initially, we have that C is a reachable configuration (for level $\log n$) and all valid configurations $S^\star$ for level $i-1$ are reachable if there is a reachable configuration S for level i, a reachable sibling configuration S', and a strategy profile c such that $p(S,S',S^\star,c) \neq 0$.

We start by computing a minimum random game cover in $2^{O(x)} \cdot n^{O(1)}$ time (Proposition 2). Now we can observe that each overall outcome of the tournament is uniquely defined by the number of wins achieved by each player in the random game cover and each player in the coalition: the outcomes of each game that does not involve a player from the random game cover or a coalition player is uniquely determined. Furthermore, if we can enumerate all possible outcomes of the tournament, we can also enumerate all reachable valid configurations and all reachable valid sibling configurations. Each player in the random game cover can win at most $\log n + 1$ games. Hence, there are $O((\log n)^{|C|+x})$ possible outcomes of the tournament and hence also as many valid (sibling) configurations.

Using this, we get that the number of entries of the dynamic programming table that we need to compute in order to compute $M[\log n, C]$ is in $O((\log n)^{|C|+x+1})$. It remains to analyze the time needed to compute one entry of the table. The first sum in the recursive expression sums over all valid sibling configurations. Note that we only need to enumerate all reachable valid sibling configurations to compute the recursion. The number of reachable valid sibling configurations is in $O((\log n)^{|C|+x})$. Next, we have a maximum over all strategy profiles, of which there are $2^{|C|}$ many. Lastly, we sum over all valid configurations for the previous level. Again, it is sufficient to sum over all reachable valid sibling configurations, of which there are $O((\log n)^{|C|+x})$ many. Furthermore, the functions p (note that we overloaded notation here) can clearly be computed in polynomial time. Now we use the well-known fact that for $k \leq n$ we have $(\log n)^{O(k)} \subseteq k^{O(k)} + n^{O(1)}$ (see e.g. (Ramanujan and Szeider 2017)). It follows that the overall time required to compute all entries of M is in $(|C|+x)^{O(|C|+x)} \cdot n^{O(1)}$. □

Lastly, we observe the following: if we use the definition of the dynamic program as a recursive algorithm, that is, we recompute each value in the recursion instead of looking it up in the table, then we can solve ACCM-GKT using only polynomial space. Hence, we obtain the following.

Theorem 10. *ACCM-GKT is contained in PSPACE.*

Finally, we consider the problem of computing best responses. Notice when computing the entry $M[r, C]$ of the dynamic programming table, we have only one valid sibling configuration for L_r, say S', namely the players that are seeded as the siblings of the coalition players. Hence, we have that a best response for the first round is a strategy profile that maximizes the expression

$$\max_{\substack{\text{strategy profile} \\ c}} \left(\sum_{\substack{\text{valid configuration} \\ S^\star \text{ for } L_r}} p(C,S',S^\star,c) \cdot M[r-1,S^\star] \right).$$

It follows that the dynamic programming algorithm can also compute best responses (in the same running time bounds). Formally, we get the following.

Corollary 11. BR-ACCM-GKT *can be solved in* $n^{O(|C|)}$ *time and* BR-ACCM-GKT *can be solved in polynomial space.* BR-ACCM-KT *can be solved in* $(|C|+x)^{O(|C|+x)} \cdot n^{O(1)}$ *time, where* x *is the size of a minimum random game cover.*

6 Conclusion

Our work introduces adaptiveness to the concept of coalition manipulation in probabilistic knockout tournaments, and presents several highly non-trivial results in this regard. The same questions can be asked for the concept of budget-constrained manipulation in knockout tournaments. We conjecture that all of our hardness results extend to this setting. However, the study of positive results in this setting is left for future research. Additionally, we leave the following questions for future research:

- Is ACCM-KT PSPACE-hard?
- Do the non-generalized problems considered in this paper belong to the class FPT when parameterized by the coalition size?
- Do the non-generalized problems considered in this paper belong to the class FPT when parameterized by x?

Acknowledgments

Juhi Chaudhary was supported by the DAE, Government of India, project nr. RTI4001. Hendrik Molter was supported by the ISF, grant nr. 1470/24 and by the European Union's Horizon Europe research and innovation programme under grant agreement 949707. Meirav Zehavi was supported by the ISF, grant nr. 1470/24 and by the ERC grant nr. 101039913 (PARAPATH).

References

538. 2023. Fivethirtyeight. https://fivethirtyeight.com/. Accessed: 2025-01-16.

Arora, S.; and Barak, B. 2009. *Computational Complexity: A Modern Approach*. Cambridge University Press.

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.

Brandt, F.; and Fischer, F. 2007. PageRank as a weak tournament solution. In *Proceedings of the 3rd International Workshop on Web and Internet Economics (WINE)*, 300–305. Springer.

Chaudhary, J.; Molter, H.; and Zehavi, M. 2024a. Adaptive Manipulation for Coalitions in Knockout Tournaments. *arXiv preprint arXiv:2412.11799*.

Chaudhary, J.; Molter, H.; and Zehavi, M. 2024b. How to Make Knockout Tournaments More Popular? In *Proceedings of the 2024 AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 9582–9589.

Chaudhary, J.; Molter, H.; and Zehavi, M. 2024c. Parameterized Analysis of Bribery in Challenge the Champ Tournaments. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2704–2712.

Connolly, R.; and Rendleman, R. 2011. Tournament qualification, seeding and selection efficiency. Technical report, Technical report 2011-96, tuck school of business.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer.

Downey, R. G.; Fellows, M. R.; et al. 2013. *Fundamentals of parameterized complexity*, volume 4. Springer.

Durand, F. 2015. *Towards less manipulable voting systems*. Ph.D. thesis, Université Pierre et Marie Curie-Paris VI.

Durand, F. 2023. Coalitional manipulation of voting rules: simulations on empirical data. *Constitutional Political Economy*, 34: 390–409.

ESPN. 2023. ESPN Analytics. http://www.espn.com/analytics. Accessed: 2025-01-16.

Fellows, M. R.; Hermelin, D.; Rosamond, F.; and Vialette, S. 2009. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1): 53–61.

Feltes, T. 2013. Match fixing in western Europe. In *Match-fixing in international sports: Existing processes, law enforcement, and prevention strategies*, 15–30. Springer.

Groh, C.; Moldovanu, B.; Sela, A.; and Sunde, U. 2012. Optimal seedings in elimination tournaments. *Economic Theory*, 49: 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.

Hill, D. 2010. A critical mass of corruption: Why some football leagues have more match-fixing than others. *International Journal of Sports Marketing and Sponsorship*, 11(3): 38–52.

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.

Konicki, C.; and Williams, V. V. 2019. Bribery in balanced knockout tournaments. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2066–2068.

Laslier, J. F. 1997. *Tournament solutions and majority voting*, volume 7. Springer.

Manoli, A. E.; and Antonopoulos, G. A. 2015. 'The only game in town?': football match-fixing in Greece. *Trends in organized crime*, 18: 196–211.

Mattei, N.; Goldsmith, J.; Klapper, A.; and Mundhenk, M. 2015. On the complexity of bribery and manipulation in tournaments with uncertain information. *Journal of Applied Logic*, 13(4): 557–581.

Mattei, N.; and Walsh, T. 2016. Empirical evaluation of real world tournaments. *arXiv preprint arXiv:1608.01039*.

MediaNews. 2020b. Time out for match-fixers manipulating livestreams. https://www.europol.europa.eu/media-press/newsroom/news/time-out-for-match-fixers-manipulating-livestreams. Accessed: 2025-01-16.

MediaNews. 2023a. Most Talked-About Corruption Scandals in Sports History. https://247wallst.com/special-report/2023/09/07/17-most-talked-about-corruption-scandals-in-sports-history/. Accessed: 2025-01-16.

MediaNews. 2023c. Top Controversies of World Cup 2023. https://timesofindia.indiatimes.com/sports/cricket/icc-world-cup/news/angelo-mathews-timed-out-to-pitch-switch-top-controversies//-of-world-cup-2023/articleshow/105326878.cms. Accessed: 2025-01-16.

Predict. 2023. Predictmachine. https://www.predictionmachine.com. Accessed: 2024-01-10.

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

Rosen, S. 1985. Prizes and incentives in elimination tournaments.

Russell, T.; and Van Beek, P. 2012. Detecting manipulation in cup and round robin sports competitions. In *Proceedings of the 24th International Conference on Tools with Artificial Intelligence (ICTAI)*, volume 1, 285–290. IEEE.

Russell, T.; and Walsh, T. 2009. Manipulating tournaments in cup and round robin competitions. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory (ADT)*, 26–37. Springer.

Saarinen, S.; Goldsmith, J.; and Tovey, C. A. 2015. Probabilistic Copeland Tournaments. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1851–1852.

Schneider, J.; Schvartzman, A.; and Weinberg, S. M. 2016. Condorcet-consistent and approximately strategyproof tournament rules. *arXiv preprint arXiv:1605.09733*.

Stanton, I.; and Williams, V. V. 2013. The structure, efficacy, and manipulation of double-elimination tournaments. *Journal of Quantitative Analysis in Sports*, 9(4): 319–335.

Suksompong, W. 2021. Tournaments in Computational Social Choice: Recent Developments. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 4611–4618.

Team. 2023. Team Rankings. https://www.teamrankings.com/. Accessed: 2025-01-16.

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 Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 225–232.

Walsh, T. 2011. Where are the hard manipulation problems? *Journal of Artificial Intelligence Research*, 42: 1–29.

Williams, V. V. 2010. Fixing a tournament. In *Proceedings of the 2010 AAAI Conference on Artificial Intelligence (AAAI)*, volume 24, 895–900.

Williams, V. V. 2016. *Knockout Tournaments*, 453–474. Cambridge University Press.

Yang, Y. 2023. On the Complexity of the Two-Stage Majority Rule. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS*, 2022–2030.

Zehavi, M. 2023. Tournament fixing parameterized by feedback vertex set number is FPT. In *Proceedings of the 2023 AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, 5876–5883.