Parameterized Analysis of Bribery in Challenge the Champ Tournaments

Juhi Chaudhary [1] , **Hendrik Molter** [2] , **Meirav Zehavi** [2]
[1]School of Technology and Computer Science, TIFR, Mumbai, India
[2]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

Challenge the champ tournaments are one of the simplest forms of competition, where a (initially selected) champ is repeatedly challenged by other players. If a player beats the champ, then that player is considered the new (current) champ. Each player in the competition challenges the current champ once in a fixed order. The champ of the last round is considered the winner of the tournament. We investigate a setting where players can be bribed to lower their winning probability against the initial champ. The goal is 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. In previous work, it was shown that the problem can be solved in pseudo-polynomial time, and that it is in XP when parameterized by the number of players.

We show that the problem is weakly NP-hard and W[1]-hard when parameterized by the number of players. On the algorithmic side, we show that the problem is fixed-parameter tractable (FPT) when parameterized either by the number of different bribe values or the number of different probability values. To this end, we establish several results that are of independent interest. In particular, we show that the product knapsack problem is W[1]-hard when parameterized by the number of items in the knapsack, and that constructive bribery for cup tournaments is W[1]-hard when parameterized by the number of players. Furthermore, we present a novel way of designing mixed integer linear programs, ensuring optimal solutions where *all* variables are integers.

1 Introduction

Sports tournaments are ubiquitous at global events such as World Cups and the Olympics, national events such as sports leagues, and local events such as school competitions. While entertaining, these sports tournaments aim to impartially identify the most talented player, the *champ*, according to specific criteria. Unfortunately, the crucial requirement of ensuring fairness in this process is a highly complicated challenge. On top of the fact that every player aspires to become the champ, the ongoing monetization of sports—through advertising and lucrative brand deals awarded to winners—intensifies the competition. Accordingly, the performance of various forms of manipulation in tournaments, such as bribery, constitutes a significant body of research in social choice theory and related disciplines. These works concern, in particular, round-robin tournaments (see, e.g., [Baumeister and Hogrebe, 2021; Krumer *et al.*, 2023; Rasmussen and Trick, 2008]), cup tournaments (see, e.g., [Suksompong, 2021; Williams and Moulin, 2016; Gupta *et al.*, 2018b; Russell and Walsh, 2009; Vu *et al.*, 2009]), and challenge the champ tournaments [Mattei *et al.*, 2015].

The literature focuses on several prominent ways to manipulate a tournament. The (arguably) most natural one is to offer incentives such as bribes to specific players (individuals or part of a team), team coaches, or judges, persuading them to lose (or, in the case of judges, flip the outcome) of a match deliberately [Russell and Walsh, 2009]. We focus on the standard concept of budget-constrained bribery in tournaments [Gupta *et al.*, 2018b; Russell and Walsh, 2009; Vu *et al.*, 2009], and on challenge the champ tournaments (as well as, to some extent, cup tournaments).

Our Setting. We study the computational problem of constructive (budget-constrained) bribery in challenge the champ tournaments in the (standard) probabilistic setting, termed CONSTRUCTIVE BRIBERY FOR CHALLENGE THE CHAMP TOURNAMENTS (CBCCT). The study of the complexity of this problem was initiated by Mattei *et al.* [2015]. Challenge the champ tournaments consist of a set of $n + 1$ players, $\{e_1, \ldots, e_n, e^*\}$, where e^* is the initial champ. The (initially selected) champ e^* is repeatedly challenged by the other players. If a player beats the champ, then that player is considered the new (current) champ. Each player in the competition challenges the current champ once in the fixed order $e_1, e_2, \ldots, e_n$. The champ of the last round is considered the winner of the tournament. When we consider the possibility of manipulation in tournaments, we are supposed to possess information about the probabilities of the outcomes of the matchings. Here, the standard probabilistic model is to assume that for each pair of players that can potentially compete against each other, we know the probability of one of them beating the other (and, hence, we also know the probability of the other beating the first); see, e.g., [Mattei *et al.*, 2015; Kim and Williams, 2015; Vu *et al.*, 2008; Aziz *et al.*, 2014; Stanton and Williams, 2011]. Constructive bribery is the most ubiquitous form of manipulation in both competition and voting [Mattei *et al.*, 2015; Saarinen *et al.*, 2015; Faliszewski *et al.*, 2006; Faliszewski *et al.*, 2009; Karia *et al.*, 2023; Tao *et al.*, 2023], and its objective is to manipulate the selection process so that our favorite player/candidate wins. Here, we are often supposed to have a price associated with each possible bribing action along with a budget.

Accordingly, in CBCCT we are given, along with player set $\{e_1, \ldots, e_n, e^*\}$:

- For every player e_i, a *bribe vector*, which is a vector of price-probability pairs; each pair specifies the price of the bribe(s) required to make e_i lose against e^* with the specified probability. We can suppose that the vector includes a pair with price 0, which corresponds to the probability of e_i losing when no bribe is involved.
- A budget $B \in \mathbb{N}$.
- A threshold probability $t \in [0, 1]$.

The reason behind having a vector with more than two entries (and, in particular, losing probabilities other than 1 when a bribe is involved) is that various ways can affect the probability of a team or player losing, each having a different price. For example, we can bribe a different number of players in e_i (when e_i is a team), different coaches of e_i, the judge(s) of that specific match, alter various environmental conditions (e.g., which player plays in which court), and more.

The goal of CBCCT (formally defined in Section 2) is to determine whether the probability of e^* winning the tournament can be increased to or above t using bribes for the matches between e^* and $e_1, \ldots, e_n$ based on their respective bribe vectors, without exceeding the budget B. We remark that our model is slightly more general than the one of Mattei *et al.* [2015], since they require the probabilities to be encoded in unary and we do not.

The initial work of Mattei *et al.* [2015] proved the following results related to CBCCT:

- CBCCT belongs to NP. This follows from [Mattei *et al.*, 2015, Corollary 4.4].
- CBCCT is in XP[1] when parameterized by the number of players. This result implicitly follows from [Mattei *et al.*, 2015, Theorem 4.9 & Corollary 4.10], since the number of rounds of the tournament and the number of games in every round are upper-bounded by the number n of players.
- CBCCT can be solved in $O(B^2 n)$ time [Mattei *et al.*, 2015, Theorem 4.13], showing that CBCCT is solvable in pseudo-polynomial time.
- They established (weak) NP-hardness of a variant of CBCCT, where all probabilities are expressed as (negative) powers of two [Mattei *et al.*, 2015, Theorem 4.17]. Note that their reduction requires a compact representation of the probabilities. Hence, the problem they addressed is not a special case of CBCCT, and their reduction does not imply (weak) NP-hardness of CBCCT.

[1]We use the standard terminology in parameterized complexity [Downey *et al.*, 2013; Cygan *et al.*, 2015; Niedermeier, 2006]. An overview of the concepts used in this work is given in the full version [Chaudhary *et al.*, 2024].

Cup tournaments are extremely popular in sports competitions [Suksompong, 2021; Williams and Moulin, 2016; Vu *et al.*, 2008; Gupta *et al.*, 2018b; Manurangsi and Suksompong, 2023], voting [Vu *et al.*, 2009; Laslier, 1997], and decision making [Brandt and Fischer, 2007; Rosen, 1985]. Roughly speaking, a cup tournament is conducted in $\log_2 n$ rounds: in each round, the remaining players are paired up into matches, and the losers are knocked out of the tournament; when a single player remains, it is declared the winner. (Due to space constraints, a formal definition is given in the full version [Chaudhary *et al.*, 2024].) Concerning CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS, Mattei et al. [Mattei *et al.*, 2015] established its classification within NP. Additionally, for the deterministic setting, they showed that CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS can be efficiently solved in polynomial time using a dynamic programming algorithm. Furthermore, they introduced a variant of CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS, termed EXACT BRIBERY, where the goal is to precisely spend a budget of B, keeping other things the same. This variant was proven to be NP-complete.

Our Contribution. We start with establishing the (classical) computational complexity of CBCCT. In Section 3 we show the following.

- CBCCT is weakly NP-hard.

This motivates developing parameterized algorithms [Downey *et al.*, 2013; Cygan *et al.*, 2015; Niedermeier, 2006] for the problem. We consider three parameters: the number of players, the number of distinct bribe values, and the number of distinct probability values.

Number of players. Tournaments often involve relatively few players. For example, usually, Tennis tournaments involve around 128 players, and boxing championships involve around 30 players in a weight category. Hence, the number of players is a highly practical parameter. However, in Section 3 we show the following

- CBCCT is W[1]-hard when parameterized by the number n of players.

This implies that the XP-algorithm by Mattei *et al.* [2015] presumably cannot be improved to an FPT-algorithm. To prove the result, we also show that the PRODUCT KNAPSACK problem is W[1]-hard when parameterized by the number of items in the knapsack. We believe this is a valuable result in its own right: PRODUCT KNAPSACK can be a useful source problem for reductions to additional problems in social choice that concern probabilities (and, hence, a product of numbers) as well. Moreover, we show that our result for CBCCT further implies that CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS is W[1]-hard, too, when parameterized by the number of players. We believe that our aforementioned implication nicely extends the results of the literature on cup tournaments, which is abundant with studies of various forms of manipulation (including bribery) from the perspective of parameterized complexity [Gupta *et al.*, 2018b; Zehavi, 2023; Konicki, 2019; Ramanujan and Szeider, 2017; Aziz *et al.*, 2014; Gupta *et al.*, 2018a; Aronshtam *et al.*, 2017].

Number of distinct bribe and probability values. The number of distinct bribes is a well-motivated parameter: often, prices of the same action do not have that many possibilities. For example, a certain judge or coach will ask for the same price (or have a small range of prices) irrespective of the player or team involved. Furthermore, it is conceivable that if, say, the budget is thousands of dollars, then bribes that are not multiplications of one thousand will not be discussed—this, too, reduces the number of distinct bribes possible. Moreover, the number of distinct probabilities is likely to be small as well. Probabilities are, essentially, rough estimations, and hence, even if we have a wide range of them, they can be rounded up to the closest value from a (predetermined) small-sized set of probabilities. In Section 4, we show that these parameters yield tractability.

- CBCCT in FPT when parameterized by the number of distinct bribe values.
- CBCCT in FPT when parameterized by the number of distinct probability values.

Both algorithms exhibit similarities and are derived through mixed integer linear program (MILP) formulations for CBCCT. To obtain the results, we develop a novel method of designing MILPs that are guaranteed to have optimal solutions where *all* variables are set to integer values. We believe this technique can be useful in various application areas and hence is of independent interest.

Application to Campaign Management. A deeper look into the definition of the CBCCT problem shows that the order of the players $e_1, e_2, \ldots, e_n$ is irrelevant to its answer—i.e., if we reorder them, we obtain an equivalent instance in terms of whether the answer to our specific objective is yes or no. (Of course, reordering might affect $e_1, e_2, \ldots, e_n$, but not e^*, who has to play and win against all of them.) Thus, we can, essentially, suppose that $e_1, e_2, \ldots, e_n$ are unordered. This gives rise to other applications of our results, e.g., to the area of *campaign management* [Bredereck *et al.*, 2016; Elkind *et al.*, 2009; Elkind and Faliszewski, 2010]. Specifically, we can think of e^* as a candidate (person, idea, or product) that aims to win the election/be approved, and of each e_i as a voter (possibly representing a group of individuals who cast a single vote), whose support/consent is essential to e^*. Then, for each e_i, a price-probability pair represents an amount of money to invest in winning e_i's support/consent (e.g., by advertising) and the estimated probability of that amount being enough.

A discussion about some other related works can be found in the full version [Chaudhary *et al.*, 2024].

2 Problem Setting and Preliminaries

In the setting of challenge the champ tournaments, we have $n + 1$ players, say, $\{e_1, \ldots, e_n, e^*\}$. Player e^* is initially the champ. In each of the n rounds, player e_i challenges the current champ and is considered the new champ if they win the challenge. We formally define a challenge the champ tournament as follows. A tournament tree for a challenge the champ tournament is visualized in Fig. 1.

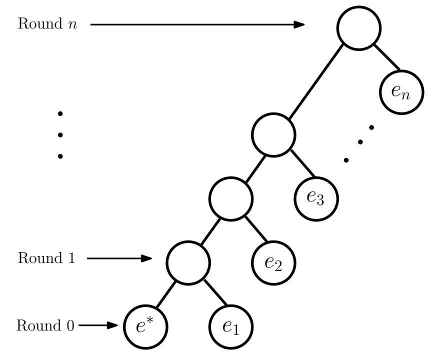

Figure 1: Illustration of a challenge the champ tournament with players $\{e_1, \ldots, e_n, e^*\}$. Here, e^* is the initial champ.

Definition 1 (Challenge the Champ Tournament)**.** *A challenge the champ tournament consists of a set of $n+1$ players $\{e_1, \ldots, e_n, e^*\}$ and has n rounds. Initially, player e^* is considered the champ (of round 0). In round $i > 0$, player e_i challenges the champ of round $i-1$, say player e. If e_i beats e, then e_i becomes the champ of round i. Otherwise, e is the champ of round i. The champ of round n is considered the* winner *of the tournament.*

Constructive Bribery. In this paper, we investigate the setting where players from the set $\{e_1, \ldots, e_n\}$ can be influenced through bribes to reduce their chances of winning against e^*. Our objective is to determine if a specific budget for bribes can be allocated in such a way that player e^* maintains the champ title and wins the tournament with a predefined probability, called *threshold value t*, after facing each player in $\{e_1, \ldots, e_n\}$. Note that since we are only interested in cases where e^* wins all games, the round in which each challenger plays against the champ does not matter.

To formalize the problem, we introduce a so-called *bribe vector* for each player in $\{e_1, \ldots, e_n\}$. We denote by C_i the bribe vector for player e_i. Intuitively, C_i specifies how much it costs to bribe players into lowering their winning probability against e^*.

Definition 2 (Bribe Vector)**.** *Let $e_i \in \{e_1, \ldots, e_n\}$ be a player and let $\ell_i \in \mathbb{N}$ be the number of different bribes that e_i accepts. Then, the* bribe vector *$C_i \in (\mathbb{N} \times [0,1])^{\ell_i}$ is a vector of length ℓ_i with elements from $\mathbb{N} \times [0,1]$. Each element $C_i[j] = (b_j, p_j) \in \mathbb{N} \times [0,1]$ implies that bribing player e_i with amount b_j increases their losing probability when playing against e^* to p_j. We call b_j a* bribe value *and p_j a* probability value. *Furthermore, we require for all i that $C_i[1] = (0, p_1)$, that is, the first entry of each bribe vector contains the losing probability when playing against e^* when no bribes are used. Moreover, we require that if $j < j'$ then $b_j < b_{j'}$, where $C_i[j] = (b_j, p_j)$ and $C_i[j'] = (b_{j'}, p_{j'})$.*

Now, given a set of bribes $\{j_1, \ldots, j_n\}$ for the players $\{e_1, \ldots, e_n\}$ (where $j_i = 1$ if player e_i is not bribed), the total cost of the bribes is $\sum_i b_{j_i}$, and the winning probability of e^* is $\prod_i p_{j_i}$, where $(b_{j_i}, p_{j_i}) = C_i[j_i]$. The main problem that we study in this paper is formally defined as follows.

CONSTRUCTIVE BRIBERY FOR CHALLENGE THE CHAMP TOURNAMENTS (CBCCT)

Input: A set of players $P = \{e_1, \ldots, e_n, e^*\}$, a bribe vector $C_i \in (\mathbb{N} \times [0,1])^{\ell_i}$ for each player $e_i \in P$, a probability threshold $t \in [0,1]$, and a budget $B \in \mathbb{N}$.

Question: Can we raise e^*'s probability of winning the challenge the champ tournament to at least t by bribing the players $\{e_1, \ldots, e_n\}$ according to their bribe vectors and not exceeding the budget B?

Furthermore, we make the following observation about bribe vectors. We call a bribe vector C *monotone*, if for all $j < j'$ we have that $p_j < p_{j'}$, where $C[j] = (b_j, p_j)$ and $C[j'] = (b_{j'}, p_{j'})$.

Lemma 1. *Given an instance of CBCCT with bribe vectors C_i with $i \in \{1, \ldots, n\}$, we can compute an equivalent instance in polynomial time with monotone bribe vectors.*

Proof. Assume that there is a player $e_i \in \{e_1, \ldots, e_n\}$ such that C_i is not monotone, that is, for some $1 \leq j < j' \leq |C_i|$ we have $p_j \geq p_{j'}$, where $C_i[j] = (b_j, p_j)$ and $C_i[j'] = (b_{j'}, p_{j'})$. Let j, j' be such that $j' - j$ is minimal. Note that we must have that $j = j' - 1$. Then, we create a bribe vector C'_i of length $|C_i| - 1$ such that $C'_i[\ell] = C_i[\ell]$ for all $1 \leq \ell \leq j$ and $C'_i[\ell] = C_i[\ell + 1]$ for all $j < \ell < |C_i|$.

We have that the CBCCT instance with the modified bribe vector is a yes-instance if and only if the original instance is a yes-instance. Let $C_i[j] = (b_j, p_j)$ and $C_i[j'] = (b_{j'}, p_{j'})$. If there is a solution to the original instance that uses value b_j to bribe player e_i to have losing probability p_j, then this is also a valid solution to the modified instance. If there is a solution to the original instance that uses value $b_{j'}$ to bribe player e_i to have losing probability $p_{j'}$, then we can create a valid solution to the modified instance by bribing player e_i with value b_j to have losing probability p_j. Since $b_j < b_{j'}$ and $p_j \geq p_{j'}$, the budget is not violated and the winning probability of e^* is not decreased. If there is a solution to the modified instance, then this solution is clearly also valid for the original instance.

By repeating the described procedure, we can create modified bribe vectors with the property such that the CBCCT instance with the modified bribe vectors is a yes-instance if and only if the original instance is a yes-instance. □

Due to Lemma 1, we will assume without loss of generality that the bribe vectors of all CBCCT instances are monotone.

We have expanded the concept of constructive bribery to another well-known tournament, called *cup tournament* (also known as *knockout tournaments*). Here also, the objective is to manipulate the players by offering bribes, under a given budget. Based on their assigned bribe vectors, the players can decrease their winning probability, ensuring that a designated favorite player, say e^*, emerges as the winner with a probability of at least a given threshold. It is essential to note that, unlike in CBCCT, critical matches can occur among players who are not favorites. Consequently, the bribe vectors are defined not only in relation to the favorite player but also among the players themselves. Owing to space constraints, the formal definition of CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS is relegated to the full version [Chaudhary *et al.*, 2024].

3 Hardness Results

Here, we present our computational hardness results. In particular, we show that CBCCT is weakly NP-hard and W[1]-hard when parameterized by the number of players. In particular, our results imply that the XP-algorithm for CBCCT parameterized by the number of players by Mattei *et al.* [2015] presumably cannot be improved to an FPT-algorithm.

Parameterized Hardness of Product Knapsack. PRODUCT KNAPSACK is known to be weakly NP-hard [Pferschy *et al.*, 2021; Halman *et al.*, 2019]. We show that PRODUCT KNAPSACK is W[1]-hard when parameterized by the number of items in the knapsack. This result is of independent interest and allows us to obtain other parameterized hardness results.

Theorem 1. PRODUCT KNAPSACK *is W[1]-hard when parameterized by the number of items in the knapsack.*

To prove Theorem 1, we adapt the reduction used by Halman *et al.* [2019] to establish the weak NP-hardness of PRODUCT KNAPSACK. Due to space constraints, the proof details are deferred to the full version [Chaudhary *et al.*, 2024].

In the multicolored version of PRODUCT KNAPSACK, each item is assigned to a specific color class, and the objective is to select precisely one item from each color class to fill our knapsack. More formally, it is defined as follows.

MULTICOLORED PRODUCT KNAPSACK

Input: Items $j \in N := \{1, \ldots, n\}$ with weights $w_j \in \mathbb{N}$ and profits $v_j \in \mathbb{N}$, a partition of N into $k (\in \mathbb{N})$ sets $X_1, \ldots, X_k$, a positive knapsack capacity $C \in \mathbb{N}$, and a value $V \in \mathbb{N}$.

Question: Does there exist a subset $S \subseteq N$ containing exactly one item from each X_i with $\sum_{j \in S} w_j \leq C$ such that $\prod_{j \in S} v_j \geq V$?

In the realm of parameterized complexity, when a problem is parameterized by the solution size, it is well-known that by applying the color-coding technique (see [Cygan *et al.*, 2015]), we can obtain a parameterized reduction (one-to-many) from the original problem to its multicolored counterpart parameterized by the number of colors. Thus, through a straightforward parameterized reduction, we get the following corollary of Theorem 1.

Corollary 1. MULTICOLORED PRODUCT KNAPSACK *is W[1]-hard when parameterized by the number of colors.*

Hardness of CBCCT. Using the previous results, in particular, Corollary 1, we now proceed to establish our main hardness result.

Theorem 2. *CBCCT is weakly NP-hard and W[1]-hard when parameterized by the number of players.*

Proof. We present a parameterized polynomial-time reduction from MULTICOLORED PRODUCT KNAPSACK parameterized by the number of colors to CBCCT parameterized by the number of players. By Corollary 1, we have that MULTICOLORED PRODUCT KNAPSACK is W[1]-hard when parameterized by the number of colors.

Let $\psi = (X_1, \ldots, X_k, C, V)$ be a given instance of MULTICOLORED PRODUCT KNAPSACK, where $v^i_1, \ldots, v^i_{|X_i|}$ denote the profits and $w^i_1, \ldots, w^i_{|X_i|}$ denote the weights of the items present in the color class X_i for every $i \in [k]$. Here, we assume without loss of generality that $w^i_1 \leq w^i_2 \leq \cdots \leq w^i_{|X_i|}$. Now, we construct an instance $\phi = (P, \overline{C}, t, B)$ of CBCCT with $k+1$ players, where $P = \{e_1, \ldots, e_k, e^*\}$ and $\overline{C}$ contains the bribe vectors for every player in $P \setminus \{e^*\}$, as follows:

Player e^* is the initial champ. Here, we informally say that for each $i \in [k]$, the player e_i corresponds to the color class X_i. Also, for each $i \in [k]$, we set the bribe vector C_i corresponding to the items in the color class X_i. Formally, the jth entry of the vector C_i is defined as $C_i[j] = (w^i_j, v^i_j)$. Here, note that the length of the vector C_i is $|X_i|$. Finally, we set budget $B = C$, and we set the threshold value $t = V$.

This finishes the construction, which can clearly be done in polynomial time. Note that the number of players in the constructed instance is $k+1$. Now, we claim that ψ is a yes-instance of MULTICOLORED PRODUCT KNAPSACK if and only if ϕ is a yes-instance of CBCCT.

($\Rightarrow$): Assume that ψ is a yes-instance of MULTICOLORED PRODUCT KNAPSACK. Let S be a solution of ψ such that $\sum_{j \in S} w_j \leq C$ and $\prod_{j \in S} v_j \geq V$. Now, we bribe the players in $P \setminus \{e^*\}$ as follows. Without loss of generality, assume that w_j and v_j correspond to the weight and price of the item in the solution that is present in the color class X_j. Then by construction, we have that for player $e_j \in P \setminus \{e^*\}$ there must be some entry in the bribe vector C_j that corresponds to the pair (w_j, v_j). We bribe player e_j with value w_j to lower their winning probability against e^* to v_j. By construction, we do not exceed the budget, since $C = B$. Furthermore, the product of the winning probabilities v_j of e^* against players e_j equals at least the threshold $t = V$.

($\Leftarrow$): Assume that ϕ is a yes-instance of CBCCT. Hence, there exists a set of bribes of total cost at most $B = C$, such that the probability that e^* wins the tournament is at least $t = V$. Now, for every $i \in [k]$, there must be an entry in C_i that specifies how player e_i is bribed. Let player e_i be bribed according to $C_i[j] = (w^i_j, v^i_j)$. We put the corresponding item in X_i into the knapsack. Thus, we have constructed a solution for MULTICOLORED PRODUCT KNAPSACK of total weight at most C, where the product of the values is at least V since it equals the winning probability of player e^*. □

Parameterized Hardness of Constructive Bribery for Cup Tournaments. The following result establishes that when the parameter is the number of players, the existence of an FPT algorithm remains unlikely for CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS as well.

Theorem 3. CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS *is W[1]-hard when parameterized by the number of players.*

We give a parameterized polynomial-time reduction from CBCCT to CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS, with both problems being parameterized by the number of players. Due to space constraints, the proof of Theorem 3 is relegated to the full version [Chaudhary *et al.*, 2024].

4 Algorithmic Results

In this section, we present our algorithmic results for CBCCT. We show two fixed-parameter tractability results. One is for the number of distinct bribe values as a parameter, and the other is for the number of distinct probability values as a parameter. Both algorithms are similar and are obtained by mixed integer linear program formulations for CBCCT.

MIXED INTEGER LINEAR PROGRAM (MILP)	
Input:	A vector x of n variables of which some are considered integer variables, a constraint matrix $A \in \mathbb{R}^{m \times n}$, two vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and a target value $t \in \mathbb{R}$.
Question:	Is there an assignment to the variables such that all integer variables are set to integer values, $c^\intercal x \geq t$, $Ax \leq b$, and $x \geq 0$?

Note that MILPs are also often considered to be optimization problems where instead of requiring $c^\intercal x \geq t$, the value of $c^\intercal x$ should be maximized. MILPs are known to be solvable in FPT-time when the number of integer variables is the parameter [Lenstra, 1983; Dadush *et al.*, 2011].

Theorem 4 ([Lenstra, 1983; Dadush *et al.*, 2011])**.** *MILP is FPT when parameterized by the number of integer variables.*

We build our MILP formulations in a specific way that ensures that there always exist optimal solutions where *all* variables are set to integer values. To this end, we establish a general result concerning MILPs that, to the best of our knowledge, has not been employed before. While this result can straightforwardly be derived from known results, it may be of independent interest.

Proposition 1. *Let the following be an MILP.*

$$\max c^\intercal x \text{ subject to } Ax \leq b, x \geq 0.$$

Let $x = (x_{int}\, x_{frac})^\intercal$, where x_{int} (resp. x_{frac}) denote the integer (resp. fractional) variables of the MILP. Let $A = (A_{int}\, A_{frac})$ where A_{int} are the first $|x_{int}|$ columns of A, that is, the coefficients of the integer variables, and A_{frac} are the remaining columns, that is, the coefficients of the fractional variables. If A_{frac} is totally unimodular, then there exists an optimal solution to the MILP where all variables are set to integer values.

Proof. Let $x^\star$ be an optimal solution to the MILP. Suppose that A_{frac} (as defined in Proposition 1) is totally unimodular. Let $c^\star$ denote the objective value achieved by $x^\star$. Let $x^\star_{\text{int}}$ be the assignment to the integer variables, and let $x^\star_{\text{frac}}$ be the assignment to the fractional variables of the MILP in the optimal solution $x^\star$. Let $c = (c_{\text{int}}\, c_{\text{frac}})^\intercal$, where c_{int} are the first $|x_{\text{int}}|$ entries of c, that is, the coefficients of the integer variables, and c_{frac} are the remaining entries, that is, the coefficients of the fractional variables. Define the following linear program (LP):

$$\max c_{\text{frac}}^{\mathsf{T}} x_{\text{frac}} \text{ subject to } A_{\text{frac}} x_{\text{frac}} \leq \hat{b}, x_{\text{frac}} \geq 0,$$

where $\hat{b}$ are the last $|x_{\text{frac}}|$ entries of $b - A_{\text{int}} x_{\text{int}}^{\star}$. Clearly, we have that $x_{\text{frac}}^{\star}$ is a feasible solution to the LP that achieves objective value $c_{\text{frac}}^{\star} = c^{\star} - c_{\text{int}}^{\mathsf{T}} x_{\text{int}}^{\star}$.

It is well known that since $\hat{A}$ is totally unimodular, the LP admits an optimal solution where all variables are set to integer values [Dantzig, 1956]. Let $x_{\text{frac}}^{\star\star}$ denote an optimal solution for the LP that sets all variables to integer values. Clearly, the achieved objective value of the solution $x_{\text{frac}}^{\star\star}$ is at least $c_{\text{frac}}^{\star}$.

Now, if we set the fractional variables of the MILP to $x_{\text{frac}}^{\star\star}$ (instead of $x_{\text{frac}}^{\star}$) and set the integer variables of the MILP to $x_{\text{int}}^{\star}$, we obtain a feasible solution to the MILP that achieves objective value at least $c^{\star}$. It is feasible since otherwise, a constraint in the LP must be violated. It has an objective value of at least $c^{\star}$ since the objective value achieved in the LP is at least $c_{\text{frac}}^{\star} = c^{\star} - c_{\text{int}}^{\mathsf{T}} x_{\text{int}}^{\star}$. The objective value achieved by the newly constructed solution to the MILP is also at most $c^{\star}$, since $c^{\star}$ is the objective value achieved by an optimal solution. We conclude that there exists an optimal solution to the MILP that sets all variables to integer values. □

Now, we are ready to state our results.

Theorem 5. *CBCCT is FPT when parameterized by the number of distinct bribe values.*

Theorem 6. *CBCCT is fixed-parameter tractable when parameterized by the number of distinct probability values.*

Due to space constraints, we only give a proof of Theorem 5, and the proof of Theorem 6 can be found in the full version [Chaudhary *et al.*, 2024]. To prove Theorem 5, we provide a mixed integer linear program (MILP) formulation for CBCCT where the number of integer variables is upper-bounded by a function of the number of distinct bribe values in the CBCCT instance.

Proof of Theorem 5. We provide the MILP formulation for CBCCT where the number of integer variables is upper-bounded by a function of the number of distinct bribe values. Assume we are given an instance of CBCCT. We construct an MILP as follows.

Let $v_{\#}$ denote the number of distinct bribe values, and V denote the set of distinct bribe values. For each combination of a subset $V' \subseteq V$ and a value in that subset $v' \in V'$, we create an integer variable $x_{v',V'}$ that, intuitively, counts how many times we bribe a player with value v' that has set of bribe values V'. We call the set of probability values P in the bribe vector of a player e_i the player's *probability profile*. From Lemma 1 follows that if two players e_i, e_j have the same probability profile P and have the same set of bribe values V', we must have that $|P| = |V'|$ and hence bribing e_i with some value $v' \in V'$ and bribing e_j with the same v' increases the losing probability of e_i and e_j to the same $p \in P$. We denote this probability with $p = p(P, v', V')$. In other words, players are uniquely characterized by their probability profile and their set of bribe values.

For each combination of a subset $V' \subseteq V$, a value in that subset $v' \in V'$, and a probability profile P (that appears in the CBCCT instance), we create a rational-valued variable $x_{P,v',V'}$ that, intuitively, counts how many times a player that has set of bribe values V' and probability profile P is bribed with value v' (to increase its losing probability to a uniquely determined $p = p(P, v', V')$).

We want to maximize the following.

$$\prod_p p^{\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'}}$$

This is equivalent to maximizing the logarithm of the expression. Hence, we have the following (linear) objective function.

$$\sum_p \log p \left(\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'} \right)$$

We have the following constraints. The first one ensures that we do not violate the budget.

$$\sum_{v',V'} v' \cdot x_{v',V'} \leq B \tag{1}$$

The second set of constraints ensures that the number of times we use a value v' to bribe a player that has the set of bribe values V' (which is specified by $x_{v',V'}$) is the same as the sum of all times we use value v' to bribe a player that has set of bribe values V' and that has probability profile P.

$$\forall\, v', V' \colon \sum_P x_{P,v',V'} = x_{v',V'} \tag{2}$$

The third set of constraints ensures that we do not use a value v' to bribe a player that has the set of bribe values V' and probability profile P too many times. Let $n_{P,V'}$ denote the number of players that have a set of bribe values V' and the probability profile P.

$$\forall\, P, V' \colon \sum_{v'} x_{P,v',V'} = n_{P,V'} \tag{3}$$

Lastly, in the fourth set of constraints, we require that all fractional variables $x_{P,v',V'}$ are non-negative.

$$\forall\, P, v', V' \colon 0 \leq x_{P,v',V'} \tag{4}$$

It is easy to observe that the overall number of variables and constraints is in $2^{O(v_{\#})} \cdot n$ whereas the number of integer variables is in $2^{O(v_{\#})}$. By Theorem 4, we can compute an optimal solution for the MILP in FPT-time with respect to the number $v_{\#}$ of distinct bribe values.

In the remainder, we show that there is a solution to the MILP with

$$\prod_p p^{\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'}} \geq t$$

if and only if the input instance of CBCCT is a yes-instance.

($\Rightarrow$): Assume the input instance of CBCCT is a yes-instance. Then, it is possible to bribe players using budget B such that the winning probability of e^* is at least t. Let player e_i be bribed with value v_i in the solution and let the resulting losing probability of e_i versus e^* be p_i.

We construct a solution for the constructed MILP as follows. Initially, we set all variables to 0. Now iterate over all players. Let player e_i have a set of bribe values V' and probability profile P. Then we increase the value of variable $x_{P,v_i,V'}$ by 1. Note that after his procedure, the constraints (3) and (4) are clearly met.

Afterward, we set all integer values such that constraints (2) are satisfied. Since these are equality constraints, this uniquely specifies how the integer variables are set. Furthermore, since we set all fractional variables to integer values in the previous step, we clearly also set all integer variables to integer values.

Next, we argue that constraint (1) is satisfied. To this end, note that every bribe with value v' is accounted for exactly once by increasing the value of a variable $x_{P,v',V'}$ by 1. It follows that the same bribe is accounted for exactly once in the value of variable $x_{v',V'}$. Since the input instance is a yes-instance, the sum of all bribes is at most B, hence, we have that constraint (1) is satisfied.

Lastly, we argue that

$$\prod_p p^{\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'}} \geq t.$$

Similarly as in the argument before, note that every challenge that player e^* can win with probability p is accounted for exactly once by increasing the value of a variable $x_{P,v',V'}$ such that $p = p(P,v',V')$ by 1. Hence, the number of challenges that e^* can win with probability p is $\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'}$. Since the input instance is a yes-instance, player e^* can win all challenges with a probability of at least t. It follows that the above inequality is fulfilled.

($\Leftarrow$): Assume that we have a solution $x^\star$ to the created MILP instance such that

$$\prod_p p^{\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'}} \geq t.$$

In the following, we show how to bribe the players to increase the overall winning probability of e^* to at least t.

To this end, we show that there exists an optimal solution to the created MILP where *all* variables are set to integer values. We do this using Proposition 1.

Note that constraint (1) is independent of the fractional variables. Furthermore, constraints (3) are independent from the integer variables. We transform constraints (2) to constraints for the fractional variables by treating the integer variables as arbitrary constants. After that, we have a constraint matrix for the fractional variables consisting of the modified constraints (2) and constraints (3). In the following, we show that the corresponding constraint matrix is totally unimodular, which then by Proposition 1 implies that there exists an optimal solution to the MILP that sets all variables to integers.

First, note that the constraints (2) partition the set of fractional variables, that is, each fractional variable is part of exactly one of the constraints (2). We have the same for the constraints (3). Furthermore, the coefficients in the constraint matrix for each variable are either 1 (if they are part of a constraint) or 0. It follows that the constraint matrix is a 0-1 matrix with exactly two 1's in every column. Additionally, in each column, we have that one of the two 1's appears in a row corresponding to the constraints (2) and the other 1 is in a row corresponding to the constraints (3). This is a sufficient condition for the constraint matrix to be totally unimodular [Dantzig, 1956].

Thus, from now on, we can assume that the optimal solution $x^\star$ to the MILP sets all variables to integer values. We construct the bribes for the players as follows. Let V' be a set of bribe values and P be a probability profile. Consider the set $E_{P,V'}$ of all players that have the set of bribe values V' and that have probability profile P. For each $v' \in V'$ we bribe $x_{P,v',V'}$ players of $E_{P,V'}$ with v'. Note that constraints (3) ensure that there are sufficiently many players in $E_{P,V'}$. Each bribe done this way is accounted for exactly once in the variable $x_{v',V'}$ due to the constraints (2). Since constraint (1) is satisfied, we have that the total amount of bribes does not exceed the budget B.

It remains to show that the winning probability of e^* after bribing the players is at least t. To this end, recall that a player with the set of bribe values V' and probability profile P loses with probability p when playing against e^* if and only if they are bribed with some $v' \in V'$ such that $p = p(P,v',V')$. It follows that the number of players that have losing probability p when playing against e^* is

$$\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'}.$$

Hence, we have that the overall winning probability of e^* is

$$\prod_p p^{\sum_{P,v',V'|p=p(P,v',V')} x_{P,v',V'}},$$

which by assumption is at least t. $\square$

5 Conclusion

In our work, we investigated the parameterized complexity of CBCCT, a natural tournament bribery problem. There are several natural directions for future research. It remains open whether CBCCT is NP-hard when the probabilities are encoded in unary and bribe values are encoded in binary (this question has already been raised by Mattei *et al.* [2015]). In fact, it is open whether PRODUCT KNAPSACK is NP-hard when the item values are encoded in unary and item sizes are encoded in binary.

Furthermore, note that our FPT-algorithms have double-exponential running times in the parameter. We leave the question of whether this can be improved open. Moreover, exploring whether our MILP formulations for CBCCT can be extended to CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS would be interesting. The main difficulty is that we heavily exploit in our MILP formulations that the ordering of the matches (that is, the seeding) is irrelevant in CBCCT, which is not the case in CONSTRUCTIVE BRIBERY FOR CUP TOURNAMENTS.

Ethical Statement

There are no ethical issues.

Acknowledgments

Juhi Chaudhary and Meirav Zehavi are supported by the ERC, grant number 101039913. Hendrik Molter is supported by the ISF, grant number 1456/18, and the European Union's Horizon Europe research and innovation programme under grant agreement 949707.

References

[Aronshtam *et al.*, 2017] Lior Aronshtam, Havazelet Cohen, and Tammar Shrot. Tennis manipulation: can we help serena williams win another tournament? or can we control a knockout tournament with reasonable complexity? *Annals of Mathematics and Artificial Intelligence*, 80(2):153–169, 2017.

[Aziz *et al.*, 2014] Haris Aziz, Serge Gaspers, Simon Mackenzie, Nicholas Mattei, Paul Stursberg, and Toby Walsh. Fixing a balanced knockout tournament. In *Proceedings of the 2014 AAAI Conference on Artificial Intelligence (AAAI)*, volume 28, 2014.

[Baumeister and Hogrebe, 2021] Dorothea Baumeister and Tobias Alexander Hogrebe. Complexity of scheduling and predicting round-robin tournaments. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 178–186, 2021.

[Brandt and Fischer, 2007] Felix Brandt and Felix Fischer. Pagerank as a weak tournament solution. In *Proceedings of the 3rd International Workshop on Web and Internet Economics (WINE)*, pages 300–305. Springer, 2007.

[Bredereck *et al.*, 2016] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Large-scale election campaigns: Combinatorial shift bribery. *Journal of Artificial Intelligence Research*, 55:603–652, 2016.

[Chaudhary *et al.*, 2024] Juhi Chaudhary, Hendrik Molter, and Meirav Zehavi. Parameterized analysis of bribery in challenge the champ tournaments. *CoRR*, abs/2403.17587, 2024.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Dadush *et al.*, 2011] Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative lattice algorithms in any norm via m-ellipsoid coverings. In *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 580–589. IEEE, 2011.

[Dantzig, 1956] George Bernard Dantzig. *Linear inequalities and related systems*. Number 38. Princeton University Press, 1956.

[Downey *et al.*, 2013] Rodney G Downey, Michael R Fellows, et al. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

[Elkind and Faliszewski, 2010] Edith Elkind and Piotr Faliszewski. Approximation algorithms for campaign management. In *Proceedings to the 6th International Workshop on Internet and Network Economics (WINE)*, pages 473–482. Springer, 2010.

[Elkind *et al.*, 2009] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium of Algorithmic Game Theory (SAGT)*, pages 299–310. Springer, 2009.

[Faliszewski *et al.*, 2006] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the 2006 AAAI Conference on Artificial Intelligence (AAAI)*, volume 6, pages 641–646, 2006.

[Faliszewski *et al.*, 2009] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[Gupta *et al.*, 2018a] Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. When rigging a tournament, let greediness blind you. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 275–281, 2018.

[Gupta *et al.*, 2018b] Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. Winning a tournament by any means necessary. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 282–288, 2018.

[Halman *et al.*, 2019] Nir Halman, Mikhail Y Kovalyov, Alain Quilliot, Dvir Shabtay, and Moshe Zofi. Bi-criteria path problem with minimum length and maximum survival probability. *OR Spectrum*, 41:469–489, 2019.

[Karia *et al.*, 2023] Neel Karia, Faraaz Mallick, and Palash Dey. How hard is safe bribery? *Theoretical Computer Science*, 979:114156, 2023.

[Kim and Williams, 2015] Michael P Kim and Virginia Vassilevska Williams. Fixing tournaments for kings, chokers, and more. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 561–567, 2015.

[Konicki, 2019] Christine Konicki. *Topics in fixing knockout tournaments: bribery, NP-hardness, and parameterization*. PhD thesis, Massachusetts Institute of Technology, 2019.

[Krumer *et al.*, 2023] Alex Krumer, Reut Megidish, and Aner Sela. Strategic manipulations in round-robin tournaments. *Mathematical Social Sciences*, 122:50–57, 2023.

[Laslier, 1997] Jean François Laslier. *Tournament solutions and majority voting*, volume 7. Springer, 1997.

[Lenstra, 1983] Hendrik W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[Manurangsi and Suksompong, 2023] Pasin Manurangsi and Warut Suksompong. Fixing knockout tournaments with seeds. *Discrete Applied Mathematics*, 339:21–35, 2023.

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

[Niedermeier, 2006] Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31. OUP Oxford, 2006.

[Pferschy *et al.*, 2021] Ulrich Pferschy, Joachim Schauer, and Clemens Thielen. Approximating the product knapsack problem. *Optimization Letters*, 15(8):2529–2540, 2021.

[Ramanujan and Szeider, 2017] M Ramanujan and Stefan Szeider. Rigging nearly acyclic tournaments is fixed-parameter tractable. In *Proceedings of the 2017 AAAI Conference on Artificial Intelligence (AAAI)*, volume 31, 2017.

[Rasmussen and Trick, 2008] Rasmus V Rasmussen and Michael A Trick. Round robin scheduling–a survey. *European Journal of Operational Research*, 188(3):617–636, 2008.

[Rosen, 1985] Sherwin Rosen. Prizes and incentives in elimination tournaments, 1985.

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

[Saarinen *et al.*, 2015] Sam Saarinen, Judy Goldsmith, and Craig A Tovey. Probabilistic copeland tournaments. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1851–1852, 2015.

[Stanton and Williams, 2011] Isabelle Stanton and Virginia Vassilevska Williams. Manipulating single-elimination tournaments in the braverman-mossel model. In *Proceedings of the IJCAI Workshop on Social Choice and Artificial Intelligence*, volume 87, 2011.

[Suksompong, 2021] Warut Suksompong. Tournaments in computational social choice: Recent developments. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4611–4618, 2021.

[Tao *et al.*, 2023] Liangde Tao, Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, and Weidong Shi. Electoral manipulation via influence: probabilistic model. *Autonomous Agents and Multi-Agent Systems*, 37(1):18, 2023.

[Vu *et al.*, 2008] Thuc Vu, Alon Altman, and Yoav Shoham. On the agenda control problem in knockout tournaments. *Proceedings of the 2nd International Workshop on Computational Social Choice (COMSOC)*, 2008.

[Vu *et al.*, 2009] Thuc Vu, Alon Altman, and Yoav Shoham. On the complexity of schedule control problems for knockout tournaments. In *AAMAS (1)*, pages 225–232, 2009.

[Williams and Moulin, 2016] Virginia Vassilevska Williams and Hervé Moulin. *Knockout Tournaments*, page 453–474. Cambridge University Press, 2016.

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