ABSTRACT
Optimal seeding in balanced knockout tournaments has only been studied in very limited settings, for example, maximizing predictive power for up to 8 players using only the relative ranking of the players (ordinal information). We dramatically broaden the scope of the analysis along several dimensions. First, we propose a heuristic algorithm that makes use of available cardinal information and show an improvement in the optimality of the solution. Second, we address tournaments with size up to 128 players. Since the large number of distinct seedings prohibits finding the optimal solution, we introduce an innovative, provably correct upper bound on the optimal value. More interestingly, our heuristic and upper bound achieve objective values that are close to each other. This shows the upper bound and the heuristic solution both approximate well the optimal values. Last but not least, we also investigate two novel objectives: the expected strength of the winner, and the revenue of the tournament. The analysis of both objectives shows that our solution is indeed robust and effective.

Thuc Vu
Computer Science Department
Stanford University, California
thucvu@stanford.edu

Yoav Shoham
Computer Science Department
Stanford University, California
Microsoft Israel R&D Center
Herzliya Pituach, Israel
shoham@stanford.edu

Categories and Subject Descriptors
I.2.11 [Artificial Intelligence]: Multiagent Systems

General Terms
Tournament Design

Keywords
Tournament Design, Optimization, Heuristic Algorithm

1. INTRODUCTION
Tournaments play a very important role in many different social and commercial settings ranging from sporting events, elections, and patent races, to multi-agent settings such as choosing an agent most suitable for a task. A tournament can involve millions of people and billions of dollars, and yet there is no consensus on how a tournament should be organized. It is usually dependent on arbitrary decisions of the organizers, and it remains unclear why one design receives precedence over another.

To begin to address the questions in tournament design, we start out with balanced knockout tournament, one of the simplest but also most popular tournament formats. In this very familiar format, the players (whose number must be a power of two) are usually ranked prior to the tournament based on their past performance and statistics. They are then placed at the leaf nodes of a balanced binary tree accordingly to their rankings. Players who are assigned to sibling leaf nodes compete against each other in a pairwise elimination match. The winner of each match “moves up” the tree and then competes against the winner of the other branch. The player who reaches the root of the tournament tree is the winner of the tournament. The tournament organizer’s only discretion is how to arrange the players along the leaves of the tree. This is known as the seeding. As an example, we show in Figure 1 one of the seedings most commonly used in practice.

![Figure 1: A tournament with 8 players](image)

In this paper, we focus on the problem of determining an optimal seeding for a balanced knockout tournament. This seemingly simple question turns out to be surprisingly subtle and some of the answers are counter-intuitive. First of all, for the problem to be well defined, we need to be clear about the models of the players (which decide the match outcomes), the information available to the organizer, and the objective function being optimized. Since there are several choices for each of these quantities, the result is a large space of design problems.

Over the past 40 years this space has only been partially explored. Most of the previous work focused on maximizing the predictive power of the knockout tournaments for up to 8 players while only using the relative rankings of the players. We dramatically broaden the scope of the analysis along several dimensions. First, we introduce a simple heuristic method to make use of all available information (e.g., the winning probabilities between the players), and show, perhaps unsurprisingly, that this helps to improve the optimality of the solutions. We then extend the setting to tournaments with more than 8 players, and show how the heuristic can really make an impact in these settings.

The biggest challenge of going beyond 8 players is the rapid growth of the number of distinct seedings as a function of the number of players (specifically, \(O\left(\frac{n!}{2^n}\right)\)). As an example, for 16 players, this number is already \(638 \times 10^6\). The interdependencies between different parts of the tournament make it very difficult to find the optimal seedings analytically. And the number of possible seedings is prohibitively large for analyzing the optimality of any seeding empirically. To overcome this challenge, we propose an easy-to-compute upper bound of the predictive power. The upper bound is provably correct, but what makes it particularly interesting is that the values of the heuristic solution and the upper bound are close to each other. This shows that both of them approximate well the optimal solution. Using these bounds allows us to address tournaments of sizes up to 128 players.

In addition to the predictive power of the tournament, we introduce two additional objective functions: maximizing the expected value of the winner (which is different from maximizing the probability that the strongest player will win), and maximizing the revenue of the tournament, which we define in a way that correlates to the players’ strengths and competitiveness. For each of the objective functions, we propose a corresponding upper bound, and analyze its influence on the optimality of the solutions.

To explain our results we need to lay out the formal model with all its variants, which we do in the next section. In Section 3 we summarize the existing results from the literature. We present our heuristic algorithm in Section 4, describe our experiment setup in Section 5, provide our main results in Section 6 and 7, and conclude in Section 8.

2. THE MODEL

We divide our discussion of the formal setting into three parts: the player model, the objective function, and the type of solution allowed based on the information available.

2.1 Player Model

In this paper, we focus on the monotonic model, which is popular and well known in the literature (see for example [4, 6, 5, 8]). In this model, the players are numbered from 1 to \(n\) in descending order of their unknown intrinsic strengths or abilities. Only the winning probabilities between the players are known and they reflect the rankings of the players.

**Definition 1 (Monotonic Model).** Given a set of \(n\) players, the winning probabilities between the players form a matrix \(P\) such that \(p_{ij}\) denotes the probability that player \(i\) will win against player \(j\), \(\forall (i \neq j) : 1 \leq i, j \leq n\), and \(P\) satisfies the following constraints:

1. \(p_{ij} + p_{ji} = 1\)
2. \(p_{ij} \leq p_{(j+1)i}\) (which is actually implied by (1) and (2))
3. \(p_{ij} \geq p_{(j+1)i}\) (which is actually implied by (1) and (2))

In other words, the monotonic condition means it is always easier for players to win against opponents with worse rankings than the ones with better rankings. Even though this is a rather restrictive condition, many major sport tournaments do make the assumptions that players can be ranked prior to the tournament. Moreover, the winning probabilities can be obtained from past results or statistics.

Besides the winning probabilities, each player also has a certain value \(v_i\). This value can be interpreted as the strength or ability of the player, a measure of popularity, or even the size of the fan base, etc. The only restriction on the values of \(v\) is that they are also monotonic with regard to the ordering of the players, i.e., \(\forall (i < j) : v_i \geq v_j\).
2.2 Objective Functions

There are two approaches to determining the quality of a tournament seeding. The first one is axiomatic. For example, in [8], three axioms are proposed to specify what a “good” seeding should satisfy, called “Delayed Confrontation”, “Sincerity Rewarded”, and “Favoritism Minimized”. Alternatively, in [5], a “Monotonicity” property is put forward. In this paper we consider the alternative approach. Rather than placing axiomatic constraints on the seeding choice, we require that it optimizes a certain quantity.

Since the structure of the tournament is fixed as a balanced binary tree, given the seeding, the probability that each player will reach a certain stage (and eventually win the championship) can be easily calculated in $O(n^2)$.

Let $S$ be the set of all possible seeding sequences. Let $q_S(i)$ be the probability that player $i$ will win round $r$ in the tournament with the seeding $S \in S$ (note that the final round is log $n$). Let $Q_S(i)$ be the probability that $i$ will reach round $(r + 1)$, i.e., $Q_S(i) = \prod_{k=1}^{r} q_S(k)$. We consider three different objective functions:

1. MaxP: Maximizing the predictive power:
   $$\max_{S \in S} \sum_{i=1}^{n} Q_S(i)$$

2. MaxE: Maximizing the expected value of the winner:
   $$\max_{S \in S} \sum_{i=1}^{n} Q_S(i) \times v_i$$

3. MaxR: Maximizing the expected revenue of a tournament. We define the total revenue of a tournament as the sum of the revenues of all matches:
   $$\max_{S \in S} \sum_{r=1}^{\log n} \sum_{m=(i,j) \in M^r} Q^r_S(i) \times Q^r_S(j) \times \text{Rev}(m, r)$$
   where $M^r$ is the set of all possible matches that can happen in round $r$, and $\text{Rev}(m, r)$ is the revenue made by having the match $m$ in round $r$.

The first criterion, MaxP, is equivalent to maximizing the probability that the best player (player 1) will win the tournament. This has been the focus of much work (see, e.g., [4, 1, 3, 5]). MaxE has also appeared in the literature [4], while MaxR is novel.

There are many different ways to model the revenue of a single match. Here we make the following assumptions: (1) A match at a later round should generate more revenue per ticket sale; (2) A team with a higher value (strength, popularity) would attract a bigger audience; (3) A more competitive match would also attract more viewers. Based on these assumptions, we define the revenue of each match as the following:

$$\text{Rev}(m = (i, j)) = k \times [(v_i + v_j) - |P_{i} - P_{j}|]$$

Within the square brackets, the first term corresponds to the popularity of the two teams, and the second term indicates the competitiveness of the match between them. Higher values of the teams will increase the revenue of the match, while bigger gap in the winning probabilities will decrease it.

Besides MaxP, MaxE, and MaxR, there are other objective functions one might want to optimize for such as the fairness of the tournament, which can be formulated in many different ways. We leave these for future work.

2.3 Solution Types

In our work, we consider two types of seeding algorithms: Ordinal vs. Cardinal. For ordinal solutions, the tournament organizer only uses the rankings of the players. Thus these solutions are fixed seeding sequences that are applied for any ordered set of players regardless of the actual winning probabilities between them. We encode each seeding for a tournament of $n$ players by a permutation of the numbers between 1 and $n$. The $k^{th}$ number is the ranking of the player that will be placed at the $k^{th}$ leaf node of the binary tournament tree, from left to right.

We want to draw attention to two special seeding sequences. The first sequence is $S_1^n = [1, n, (n - 1), (n - 2), ..., 2]$. For example, $S_1^2 = [1, 8, 7, 6, 5, 4, 3, 2]$. Intuitively, this sequence aims to achieve MaxP by matching player 1 with the weakest player at each stage while letting strong competitors of player 1 match up with each other earlier.

The second sequence, called $S_2^n$, is formed by matching in the first stage the strongest player with the weakest player, the second strongest player with the second weakest player, and so on and so forth. For the successive stages, the subtree containing the strongest player is combined with the sub-tree with the weakest players possible, so on and so forth.

$$S_2^n = [...i, (n - i + 1), (n / 2 - i + 1), (n / 2 + i), ...]$$

An example is $S_2^8 = [1, 8, 4, 5, 2, 7, 3, 6]$ shown in Figure 1. This seeding is widely used in practice, especially in major sport tournaments. One possible rationale is that it can delay the confrontation between strong players until later rounds, and intuitively helps to improve the expected strength of the winner.

Most of the past results focus on ordinal solutions [4, 5, 8]. This type of solution provides us with a simple universal solution that we can apply to any set of ranked players without knowing the real strength of each player. This property makes it very useful in the case when the actual winning probabilities between the players are hard to obtain. Nevertheless, when these values are known, they can be used to improve the optimality of the solutions. The seedings decided based on winning probabilities and values of the players are called cardinal solutions. We will show a simple yet effective heuristic algorithm for finding cardinal solutions, and compare these two types of solution in later sections.

3. PAST RESULTS

In [4], the authors show that with monotonic model, for $n = 8$, the seeding sequence $S_1^4 = S_2^4 = (1, 4, 3, 2)$ will achieve both MaxP and MaxE. This accords with our intuition as discussed in the previous section. However, for $n = 8$, they show that $S_1^4$ is no longer the universal optimal seeding to achieve MaxP. They prove that the optimal solution can in fact be any one of the following 8 seeding sequences:

$$(1, 8, 6, 7, 2, 3, 4, 5), (1, 7, 5, 6, 2, 4, 3, 8)$$

$$(1, 8, 5, 7, 2, 3, 4, 6), (1, 8, 5, 6, 2, 4, 3, 7)$$

$$(1, 8, 5, 6, 2, 3, 4, 7), (1, 8, 5, 7, 2, 4, 3, 6)$$

$$(1, 7, 5, 6, 2, 3, 4, 8), (1, 8, 6, 7, 2, 3, 4, 5)$$

We denote this set of seeding sequences by $A$. Which se-
will attempt to improve the seeding through certain randomness. The process is composed of the call the resulting seeding HC(•).

When the algorithm is applied on the seeding with the best objective value so far will be used for the next round. Otherwise the seeding will be indeed the optimal sequence. Throughout our experiments, the average difference between the values of $S_1^n$ and the optimal values is very small. This suggests that even though $S_1^n$ is not always the optimal solution, it can in fact approximate the optimal value quite well. We have the following result.

**Theorem 1.** For a balanced knockout tournament of size 8, the difference between the values of the seeding $S_1^n$ and the optimal values is at most $\frac{1}{8}$ and this is also the best worst-case difference for all other ordinal solutions:

1. $\forall P, \forall S, Q_{P,S}^{\log n}(1) - Q_{P,S_1}^{\log n}(1) \leq \frac{1}{8}$
2. $\forall S \in \mathbb{S}, \exists P \in \mathbb{P}, S' \in \mathbb{S} : Q_{P,S}^{\log n}(1) - Q_{P,S'}^{\log n}(1) \geq \frac{1}{8}$

**Proof. Part 1:** first note that from [4], the optimal seeding can only be one of the 8 sequences in the set $A$:

- $[1,8,6,7,2,3,4,5]$
- $[1,7,5,6,2,4,3,8]$
- $[1,8,5,7,2,3,4,6]$
- $[1,8,5,6,2,4,3,7]$
- $[1,8,5,6,2,3,4,7]$
- $[1,7,5,6,2,4,3,6]$
- $[1,8,6,7,2,4,3,5]$
- $[1,8,6,7,2,4,5,3]$

Thus we only need to compare the difference between the objective values of these seedings and of $S_1^n$.

We slightly change the notation to use $q_i(S)$ to denote the probability that player $k$ will win round $r$ given the seeding $S$. The probability of player 1 winning the tournament is $Q_{P,S}^{\log n}(1) = \sum_{i=1}^{n} q_i(S)$. We can divide each seeding into two halves and compare each half separately. The first half represents the first two rounds of player 1, and the second half the last round.

There are four different seedings possible for the first half:

- $[1,8,6,7]$
- $[1,7,5,6]$
- $[1,8,5,6]$
- $[1,8,5,7]$

It is straightforward to show that the probability of player 1 winning in [1,8,6,7] is better than the other three. For example:

$$q_4([1,8,6,7]) \times q_2([1,8,6,7]) = p_{18}(p_{14}p_{47} + p_{17}p_{76})$$
$$p_{18}(p_{16} + (p_{17} + p_{16})p_{76}) \geq p_{18}p_{16}$$
$$\geq p_{18}(p_{16} + (p_{15} + p_{16})p_{60})$$

For the second half of the seedings, we need to compare $q_k([2,3,4,5])$ to $q_k([2,3,4,6])$, and $q_k([2,4,3,6])$ with $k \geq 5$. Let’s first consider the seeding $[2,3,4,6]$. 

6. **MAXIMIZING PREDICTIVE POWER**

In this section, we discuss the results for maximizing the predictive power of a tournament. Here we focus on the two solutions: the ordinal solution $S_1^n$, and the cardinal solution $HC(S_1^n)$.

6.1 *Results for n = 8*

When $n = 8$, even though it has been shown in [4] that $S_1^n([1 8 7 6 5 4 3 2])$ is not always the optimal seeding for maximizing the predictive power, intuitively it is still a very good candidate. And indeed in our experiments, out of 1M test cases, $S_1^n$ is optimal in 99.78% of the cases. The ordinal seeding with the second highest frequency of being optimal is $[1 8 6 7 2 4 3 5]$ but with only 0.89% of the test cases. Note that the sum of the two values is over 100% since they can be optimal simultaneously.

We can divide each seeding into two halves and compare each half separately. The first half represents the first two rounds of player 1, and the second half the last round.

There are four different seedings possible for the first half:

- $[1,8,6,7]$
- $[1,7,5,6]$
- $[1,8,5,6]$
- $[1,8,5,7]$

It is straightforward to show that the probability of player 1 winning in [1,8,6,7] is better than the other three. For example:

$$q_4([1,8,6,7]) \times q_2([1,8,6,7]) = p_{18}(p_{14}p_{47} + p_{17}p_{76})$$
$$p_{18}(p_{16} + (p_{17} + p_{16})p_{76}) \geq p_{18}p_{16}$$
$$\geq p_{18}(p_{16} + (p_{15} + p_{16})p_{60})$$

For the second half of the seedings, we need to compare $q_k([2,3,4,5])$ to $q_k([2,3,4,6])$, and $q_k([2,4,3,6])$ with $k \geq 5$. Let’s first consider the seeding $[2,3,4,6]$. 

Since \( p_{4k} \geq \frac{1}{4} (\forall k \geq 5) \), we have: \( q_1^3([2, 3, 4, k]) \leq q_1^3([2, 3, 4, 5]) \)
\[ \Rightarrow q_3^2([2, 3, 4, 5]) - q_2^2([2, 3, 4, k]) \]
\[ \leq p_{23}[p_{15}(p_{24} - p_{25}) - p_{14}(p_{24} - p_{23})] \]
\[ \leq p_{23}(p_{45} - p_{44})(p_{24} - p_{23}) \]
\[ = p_{23}(p_{45} - p_{44})(p_{24} - p_{23}) \]
\[ \leq \frac{1}{2} p_{23}(p_{45} - p_{44}) \]
Similarly: \( q_4^3([2, 3, 4, 5]) - q_3^3([2, 3, 4, k]) \leq \frac{1}{2} p_{23}(p_{45} - p_{44}) \)

We know that the sum of winning probabilities for each round must be equal to 1:
\[ \sum_{i=2,3,4,5} q_i^3([2, 3, 4, k]) = 1 = \sum_{i=2,3,4,5} q_i^3([2, 3, 4, 5]) \]

Thus we have:
\[ q_3^3([2, 3, 4, 5]) = -p_{12} \times (q_2^2([2, 3, 4, 5]) - q_2^2([2, 3, 4, k])) \]
\[ -p_{13} \times (q_2^2([2, 3, 4, 5]) - q_2^2([2, 3, 4, k])) + p_{14} \times (q_2^2([2, 3, 4, 5]) - q_2^2([2, 3, 4, k])) \]
\[ + p_{16} \times q_2^2([2, 3, 4, k]) - p_{15} \times q_2^2([2, 3, 4, 5]) \]
\[ = (p_{14} - p_{12}) \times (q_2^2([2, 3, 4, 5]) - q_2^2([2, 3, 4, k])) \]
\[ + (p_{14} - p_{13}) \times (q_2^2([2, 3, 4, 5]) - q_2^2([2, 3, 4, k])) \]
\[ + (p_{15} - p_{14}) \times q_2^2([2, 3, 4, 5]) \]
\[ + (p_{14} - p_{15}) \times q_2^2([2, 3, 4, k]) \]
\[ \leq \frac{1}{2} \frac{1}{2} p_{23}(p_{45} - p_{44}) + \frac{1}{2} p_{22}(p_{45} - p_{44}) \]
\[ + (p_{14} - p_{13})(1 - p_{45})(p_{22}p_{23} + p_{34}p_{32}) \]
\[ \leq \frac{1}{4} (p_{45} - p_{44}) + \frac{1}{4} (1 - p_{45}) \]
\[ \leq \frac{1}{4} \frac{1}{4} p_{45} \leq \frac{1}{8} \]

Now let's consider the seeding \([2,4,3,k]\), we have:
\[ q_3^2([2, 4, 3, k]) \leq q_3^2([2, 3, 4, 5]) \]
\[ q_4^2([2, 4, 3, k]) \leq q_4^2([2, 3, 4, 5]) \]
\[ q_2^2([2, 4, 3, k]) \leq q_2^2([2, 3, 4, 5]) \]

Thus the only way for the seeding \([2,4,3,k]\) to improve the winning probability of player 1 is to decrease probability of player 2 winning the second round. Let's consider the difference between the two winning probabilities:
\[ q_2^2([2, 3, 4, 5]) - q_2^2([2, 4, 3, k]) = p_{23}(p_{45} + p_{45}p_{34}) - p_{23}(p_{43}p_{45} + p_{34}p_{35}) = (*) \]

Since \( p_{23} \leq p_{23}p_{45} + p_{34}p_{35} \) then:
\[ (*) \leq p_{23}(p_{45} + p_{45}p_{34} - p_{23}) \]
\[ \leq p_{23}(p_{25} - p_{23}) \leq p_{23}(1 - p_{23}) \leq \frac{1}{4} \]
\[ \Rightarrow q_1^2([2, 3, 4, k]) - q_1^2([2, 3, 4, 5]) \leq (p_{12} - p_{13}) + (q_2^2([2, 4, 3, k]) - q_2^2([2, 3, 4, 5])) \leq \frac{1}{5} \]

Putting the two halves together, we have the desired property since the probability of player 1 winning the first two rounds (from the first half) is at most 1. This bound is tight since we can come up with an example where the equality holds.

**Part 2:** To show the bound, for each seeding, we just need to point out a winning probability matrix \( P \) and a seeding \( S' \) such that the inequality holds. Let's consider two types of seeding:

1. Player 1 plays against \([8], [7, 6], [5, 4, 3, 2]\) in round 1, 2, 3 respectively. For example: \([1 8 7 6 5 4 3 2]\).

2. All other seedings

For the first type of seeding, there are three possible choices for the second half of the seeding: \([2, 3, 4, 5]\), \([2, 3, 4, 5]\), \([2, 5, 3, 4]\). For each of the choices, we show an instance where the inequality holds:

1. For \([a_1, a_2, a_3, a_4, 2, 3, 4, 5]\): Let \( p_{12} = p_{13} = p_{23} = p_{24} = p_{43} = p_{45} = \frac{1}{2}, p_{42} = p_{45} = 1, and p_{1k} = 1(\forall k \geq 4) \). The remaining probabilities are assigned such that monotonicity holds. \([a_1, a_2, a_3, a_4, 2, 3, 4, 6]\) (in which we swap the positions of player 5 and 6 with each other) increases the winning probability of player 1 by \( \frac{1}{2} \).

2. For \([a_1, a_2, a_3, a_4, 2, 3, 4, 5, 3]\): Similar as the above.

For the second type of seeding, there must be at least one player in the seeding that is out of order, e.g., \([1 8 7 5 6 4 3 2]\) in which the player 5 is in the wrong set. Let \( k \) be that first player in the seeding starting from the left. We can use the following winning probabilities: \( p_{11} = 1(\forall i > k), p_{1i} = \frac{1}{4}(\forall i \leq k), p_{ji} = 1(\forall j < k \leq i), and p_{ji} = \frac{1}{2} \) for the remaining winning probabilities that we have not specified yet. Given a seeding of the second type, the winning probability of player 1 is at most half of \([1 8 7 6 5 4 3 2]\). Note that since \( k < 8, q_4^2(S_r^k) \geq \frac{1}{4} \) and hence the difference is at least \( \frac{1}{8} \).

Our cardinal solution \( HC(S_r^k) \), constructed by applying our heuristic function on \( S_r^k \), manages to improve the result to achieve optimality 100% of the test cases. Interestingly, \( HC(S_r^k) \) yields the same result, even though \( S_r^k \) is far from being a good candidate for maximizing predictive power. On average, \( S_r^k \) only achieves 84.12% of the optimal value. This shows that our heuristic function is robust and effective even when the initial seeding is not ideal.

### 6.2 Results for \( n > 8 \)

Since it is computationally infeasible to find the optimal solution for all the test cases, we come up with an efficient algorithm to find the upper bound of the predictive power given the winning probabilities between the players. We then compare the objective values of \( S_r^k \) and \( HC(S_r^k) \) with this upper bound.

To find an upper bound, denoted \( MaxPB \), for the objective \( MaxP \), first note that any player \( k \) in the tournament has a chance to play against player 1. For any given seeding, if player 1 and \( k \) can have a match at round \( r \), the chance of that happens is at least the lower bound of the probability that \( k \) will reach round \( r \). Moreover, given a seeding
S, at round r, there is a set $N_S$ of opponents that player 1 may match up against, i.e., the set of players in the sub-tournament tree sharing a root with player 1 at round r. The probabilities of player 1 playing against each of them sum up to 1: $\sum_{k \in N^r} P(1 \text{ plays } k) = 1$. With these observations, we calculate MaxPB as upper_bound(1 reaches round $(\log n + 1)$) using Algorithm 1.

**Algorithm 1** Calculate upper_bound$(t, r^*)$: upper bound of the probability that the player $t$ reaches round $r^*$

---

The functions `lower_bound` can be calculated in a method similar to `upper_bound` but with the order of the players reversed so that the target player $t$ will have to play against the strongest opponents first.

In a nutshell, the algorithm calculates the upper bounds of the probabilities that the player $k$ will survive each round, and then multiply those values together. For each round, the algorithm tries to match up the player against opponents as weak as possible. For example, assume we are calculating MaxPB as upper_bound(1 reaches round $(\log n + 1)$) using Algorithm 1.

**Theorem 2.** MaxPB = upper_bound(1, log $n + 1$) is the upper bound of MaxP.

**Proof.** We prove the correctness of the upper and lower bounds by using induction on the number of round $r$. We are showing the proof for the correctness of the upper bound here. The proof for the lower bound is similar.

**Base case:** When $r = 1$, the upper bound is 1, which is trivially correct.

**Inductive step:** Assume that the bounds are correct for $r$, i.e., `upper_bound(k, r)` is higher than the probability that player $k$ will reach round $r$ given any seeding, and reversely for `lower_bound(k, r)`. We need to show that they are also correct for $(r + 1)$. In order to show that the upper bound is correct, we need to show the following:

1. Given a seeding $S$, let $N_S$ be the set of players that player 1 will match up against at round $r$. We need to show that $Q_S(1) = \prod_{t=1}^{r-1} q^r_S(1) \leq \prod_{t=1}^{r-1} UB^r(k, N_S) =$ upper_bound($k, r + 1$).
2. For $S' \subseteq S$, $\prod_{t=1}^{r-1} UB^r(k, N_{S'}^r) \leq \prod_{t=1}^{r-1} UB^r(k, N_{S'}^r)$.

Part 1 is straightforward. Given a set $N_S$ of players, we know that: $\sum_{k \in N_S} P(k \text{ plays against } i) = 1$ and `upper_bound(i, r)` $\geq$ `lower_bound(i, r)`. By setting the probability of $k$ playing against all players in $N_S$ to be the lower bounds, and gradually increases these probabilities starting from the weakest players, $UB^r(k, N_S)$ is at least the probability that $k$ will survive round $r$. And by multiplying these terms together, we have the desired inequality.

Part 2 is much more complicated and we will show a sketch of the proof here. We need to show that for any given seeding $S$, using the set $N_S$ to calculate the upper bound will result in smaller value than using $N_{S'}$, in which at round $r$, $N_{S'} = \{n - 2^{-r-1} + 1 \ldots (n - 2^{-r} + 2)\}$. We show this by gradually converting $S$ to $S_r$ while not decreasing the value of the upper bound. First we note that since $S \neq S_{r+1}$, in $S$ there must exist at least one inversion: $\exists i, r_1 < r_2 : i \in N_{r_1}$ and $i + 1 \in N_{r_2}$. In other words, either $i$ or $(i + 1)$ (or both) belongs to the wrong set. Let $j$ be a player with the smallest $r_1$. If there are several such players with the same value $r_1$, we pick the player with the highest number. For example, when $S = [18546327]$, $j = 5$. We will then swap $j$ with $j + 1$, which is player 6 in this case. Assume $j \in N_{r_1}^r$, and $(j + 1) \in N_{r_2}^r$. It is easy to show that $UB^{r+1}(k, N_{r_1}^r \cup \{j + 1\}) \geq UB^{r+1}(k, N_{r_2}^r)$.

For $N_{r_1}^r \setminus \{j + 1\} \cup \{j\}$, we need to consider two cases: (1) $j$ is the player with the smallest number in $N_{r+1}$, or (2) otherwise. For the first case, one can write out the formula for $UB^{r+1}(k, N_{r_1}^r \setminus \{j + 1\} \cup \{j\}) - UB^{r+1}(k, N_{r_2}^r)$ to show that this difference is non-negative. For the second case, notice that by swapping $j$ with $(j + 1)$, the probability of $j$ playing against $k$ will be higher than the previous probability of $(j + 1)$. This means a decrease in the probability of other players $j' \neq j$ playing against $k$. Hence, this also increases the upper bound. 

In Figure 2 we show a graph plotting the experimental results for MaxP. Here we generate 100k tournaments of size $n$ for each $n \in [16, 128]$. The x-axis denotes the size of the tournament, and the y-axis denotes the average percentage of a particular solution when compared to the upper bound. As $n$ grows exponentially, the objective values of our cardinal solution remain close to the values of the upper bound. This implies that our upper bound is relatively tight, and our cardinal solution is close to being optimal. Using both of the upper bound and the cardinal solution allows us to
have a good approximation of the optimal values. With larger values of \( n \), our cardinal solution also shows a bigger improvement over the ordinal solutions (on average at least 4% improvement). This justifies the extra complexity arises from using the heuristic function.

In fact we have the following result on the lower bound of the worst case ratio of any ordinal seeding when \( n \geq 16 \).

**Theorem 3.** When \( n \geq 16 \), the ratio between the optimal value and the objective value of any ordinal solution is at least \( \frac{3n}{2(n+2)} \) in the worst case.

\[
\forall S \in \mathcal{S}, \exists P \in \mathcal{P}, S' \in \mathcal{S} : \frac{Q_{P,S'}^n(1)}{Q_{P,S}^n(1)} \geq \frac{3n}{2(n+2)}
\]

**Proof.** The proof of this theorem is similar to the proof for part 2 of Theorem 1. We consider two types of seedings:

1. Player 1 plays against players in the set \( \{n - 2^{r-1} + 1, \ldots, n - 2^r + 2\} \) at round \( r \).
2. All the other seedings

For the first type of the seeding, in the final round, player 1 will face one of the players in the set \( \{2, \ldots, \frac{n}{2} + 1\} \). In other words, the second half of the tournament tree will contain these players. Let’s consider the following winning probabilities:

- \( \forall i, j : 1 \leq i, j \leq \frac{n}{4} + 1, p_{ij} = \frac{1}{2} \)
- \( \forall i, j : 1 \leq i \leq \frac{n}{4} + 1, \frac{n}{4} + 3 \leq j \leq n, p_{ij} = 1 \)
- \( p_{1, \frac{n}{4} + 2} = 0 \)
- \( \forall i : 2 \leq i \leq \frac{n}{4} + 1, p_{i, \frac{n}{4} + 2} = \frac{1}{2} \)
- \( \forall j : \frac{n}{2} \leq j \leq n, p_{\frac{n}{4} + 2, j} = 1 \)

In other words, player (\( \frac{n}{4} + 2 \)) ties with the top (\( \frac{n}{4} + 1 \)) players except player 1, whom he loses to with probability 1. The top (\( \frac{n}{4} + 1 \)) players win with probability 1 against all other players except player (\( \frac{n}{4} + 2 \)). The probability that player (\( \frac{n}{4} + 2 \)) will get to the final is \( \frac{n}{2} \) since he has to play against another (\( \frac{n}{4} + 1 \)) players whom he ties with. In the final round, player 1 either plays against player (\( \frac{n}{4} + 2 \)) or one of the players in (\( 2, \ldots, \frac{n}{4} + 1 \)) (since they win against all the remaining players with probability 1). Thus the winning probability of player 1 with a seeding of the first type is:

\[
\frac{1 \times \frac{2}{n} + \frac{1}{2} \times \frac{n - 2}{n} = (n + 2)}{2n}
\]

One can notice that a better way to seed the players is:

\[
[\ldots(1, n), (n - 1, n - 2), (\frac{2}{n} - 1, \frac{2}{n} + 1, \ldots, \frac{2}{n} + 3), (\frac{2n}{2} - 2, \ldots, \frac{2n}{2} + 2) (\frac{2}{n} + 1, \ldots, 2)]
\]

Here we are matching up player (\( \frac{n}{4} + 2 \)) with the players in the set \( \{\frac{n}{2}, \ldots, \frac{3n}{2} - 2\} \). Since player (\( \frac{n}{4} + 2 \)) will win against these players with probability 1, he will have a probability of 1 to make to the semi-final playing against \( [\frac{2n}{2} + 1, \ldots, 2] \). Thus the chance of player (\( \frac{n}{4} + 2 \)) to get to the final is \( \frac{n}{2} \).

The winning probability of player 1 with this seeding is:

\[
\frac{1 \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} = 3}{4}
\]

Therefore the ratio of the optimal value over the objective value of any seedings of type 1 is at least \( \frac{3}{4} \times \frac{2n}{n+2} = \frac{3n}{2(n+2)} \).

For the second type of the seeding, we also use a similar strategy. Let \( k \) be the first player that is in the wrong set. We can use the following winning probabilities: \( p_{ij} = 1(\forall i < k) \), \( p_{ij} = \frac{1}{2}(\forall i \geq k) \), \( p_{ij} = 1(\forall j \geq k > i) \), and \( p_{ij} = \frac{1}{4} \) for the remaining winning probabilities. The worst-case ratio in this case is at least \( \frac{1}{2} \).

As \( n \) becomes large, the lower bound of the worst-case ratio gets close to \( \frac{3}{7} \).

![Figure 2: The average percentage of objective values of different solutions when compared to the upper bound for MaxP over 100k tournaments](image)

## 7. OTHER OBJECTIVE FUNCTIONS

In this section, we extend our experiments to address the other two objective functions: maximizing the expected value of the winner, and maximizing the expected revenue of the tournament. Specifically, we aim to analyze how the objectives influence the optimality of the ordinal and cardinal solutions.

### 7.1 Results for \( n = 8 \)

In Table 1, we present the experimental results for these two objective functions averaged over 1M tournaments. We only include here the seeding sequences that achieve optimality for at least 9% of the test cases for any objective. We also show the results for the two special seedings \( S^8_1 \) and \( S^8_2 \).

Under each objective function, the first column shows the percentage of tournaments in which the seeding achieves optimality, and the second column of the table shows the percentage of the optimal value each sequence achieves on average. The top portion contains the results for the ordinal seedings, the average results over all seedings and input, and the results of the worst seeding averaged over all input. The other portion contains the results for cardinal seedings generated by applying our heuristic function on \( S^8_1 \) and \( S^8_2 \).

For MaxE, surprisingly, the sequence [1 8 6 7 2 5 3 4] has the best chance of being optimal. This is due to the cases in which there is a big gap between player 1 and player 2. Matching up player 1 against the weakest players earlier in the tournament will guarantee that player 1 appears in the final. \( S^8_1 \), however, despite having similar structure does not perform as well since players 2 and 3 are competing against each other too early.

For the MaxR objective, \( S^8_2 \) is the best seeding. It is interesting to note that the sequence [1 8 6 7 2 5 3 4] does not...
perform well for MaxR even though it is the best seeding for MaxE. In this seeding, player 1 is matched up with all weak opponents until it reaches the final. This reduces the competitiveness of the matches and hence the revenue of the tournament. The results also show that the widely used seeding $S_2^k$ actually helps to maximize the revenue of the tournament, rather than the expected strength of the winner. This justifies the use of this seeding from the organizer’s perspective.

From the experimental data, we can make two observations: there is no ordinal solution that performs well across all three objective functions; for MaxE and MaxR, there is no seeding that achieves optimality with high frequency either. Yet, for all of the objectives, our cardinal solution almost always achieve optimality. Similar to the results for MaxP, it can also greatly improve the optimality of a given seeding even when that seeding is not the best candidate for optimality, e.g., improving $S_2^k$ from 6.87% to 94.99% for MaxE. This shows that our algorithm is very flexible and effective.

One caveat is that the actual difference in values of the top seedings compared to the optimal values is quite small. This is not true for all seedings since, for example, the worst seeding only achieves 82.88% of the optimal value on average for MaxR. The results show for the case of $n = 8$ that cardinal methods are necessary if one really cares about the objective function. However from a different perspective, if one weighs in the additional cognitive and administrative burden of cardinal methods, as well as the possible uncertainty regarding the input numbers, ordinal methods seem to provide an attractive alternative. Yet we will show in the next section how cardinal methods can make bigger improvement with larger $n$.

### 7.2 Results for $n > 8$

Once we have the bounds for the probability that any player $k$ will reach a certain round $r$, we can use them to find MaxEB, the upper bound of MaxE. Here the value of $r$ is $(\log n + 1)$, i.e., winning the tournament. If we set the winning probabilities of the stronger players to be the upper bound, and weaker players to be the lower bound such that the sum of the probabilities is 1, MaxEB is the expected value of the players based on these probabilities. We show the pseudocode for MaxEB in Algorithm 2.

#### Algorithm 2: Find MaxEB - MaxE Upper Bound

```plaintext
MaxEB = 0; sumP = 0
for k = 1 to n do
    addP ← lower_bound(k, log n + 1);
    MaxEB+ = addP × kk;
    sumP+ = addP;
end for
for k = 1 to n do
    addP ← upper_bound(k, log n + 1) - lower_bound(k, log n + 1);
    addP ← min(addP, 1 - sumP);
    MaxEB+ = addP × kk;
    sumP+ = addP;
if sumP+ ≥ 1 then
    Exit to outer loop;
end if
end for
```

To find an upper bound MaxRB of MaxR, notice that there are two components in MaxR formula: the expected value of the players and the competitiveness of each match. For the first component, we can use a similar method as in Algorithm 1 to find the upper bound on the expected value of the players at each round. For the second component, we can view each match between two players as a weighted edge in a graph with $n$ nodes. The weight of edge $(i, j)$ is the competitiveness between $i$ and $j$. Thus to find the upper bound for the second component, for each round $k$, we can find the maximum-weight matching of size $\frac{n}{2}$. This can be done in $O(n^3)$ by the algorithm described in [2] with addition of dummy nodes. The sum of the upper bounds for these two components from every round gives us MaxRB.

### Theorem 4

MaxEB and MaxRB are the upper bounds of MaxE and MaxR respectively.

The proof for both of the bounds is fairly straightforward. Due to the lack of space, we do not present it here.

In Figure 3 and Figure 4, we show the experimental results for MaxE and MaxR respectively. We compare our ordinal and cardinal solutions to the upper bounds across 100k tournaments of size $n$ for each value of $n$. The results show a similar trend here. Our cardinal solutions and the upper bounds remain close to each other as $n$ increases exponentially. However, for the MaxR objective function, the gap between them is slightly worse than with MaxP and MaxE. This is to be expected since the objective function is much more complicated here and makes it harder for the upper bound to remain tight. The improvement of the cardinal solution over the ordinal one is also smaller for MaxR. Here we optimize conflicting terms, e.g., increasing the value (strength) of one player might increase the revenue but might also decrease the competitiveness of the match if that player already has a better chance of winning. Hence the heuristic algorithm is not as effective as for the other objective functions.

### 8. CONCLUSION
We have investigated the problem of finding optimal seedings for knockout tournaments. We substantially expanded the scope of the problem by considering two types of solution, three different objective functions, and tournaments of size up to 128. Across the settings, we showed that the optimal seeding sequence can vary significantly depending on the objective and the actual winning probabilities. An ordinal solution can be a good choice for one objective but not another.

We proposed a simple and efficient heuristic algorithm for finding the optimal seeding. We introduced for each objective function a non-trivial upper bound that allowed us to perform experiments for $n$ up to 128. The results show that our heuristic can significantly improve the objective values of ordinal solutions. Moreover, the fact that our cardinal solutions approximate the upper bounds well shows that our solutions are close to optimal and our bounds are tight.

In future work, one possible extension would be to take into consideration other objective functions such as the interestingness or the fairness of the tournament. One can also use more complicated functions to better model the revenue of the tournament. Another extension is to relax the monotonicity condition of the winning probabilities between the players. Or on the other hand, one can try to figure out whether there exist PAC bounds for the values of $S_n^1$ with MaxBest objective and $S_n^2$ with MaxExp objective, and what properties the winning probabilities must hold for such bounds to exist.

9. REFERENCES