# Optimal Seeding in Knockout Tournaments*

# (Extended Abstract)


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



## 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 broaden the scope of the analysis along several dimensions: tournaments of size up to 128, different player models, ordinal as well as cardinal solutions, and two additional objective functions.



## Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Multiagent Systems

## General Terms

Economics, Algorithms, Experimentation

## 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. 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.

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 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, $O(\frac{n!}{2^{n-1}})$). 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.

## 2. THE SETTING

### 2.1 Player Model

We focus on the monotonic model, which is popular and well known in the literature. In this model, the players are numbered from 1 to $n$ in descending order of $v_i$, their unknown intrinsic strengths or abilities. Only the winning probabilities between the players are known and they reflect the ranking 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.* $p_{ij} \leq p_{i(j+1)}$

### 2.2 Objective Functions

Let $\mathbb{S}$ be the set of all possible seeding sequences. Let $q^r_S(i)$ be the probability that player $i$ will win round $r$ in the tournament with the seeding $S \in \mathbb{S}$ (note that the final round is $\log n$). Let $Q^r_S(i)$ be the probability that $i$ will reach


*This work was supported in part by NSF grant IIS-0205633-001 and in part by a BSF grant.

**Cite as:** Optimal Seeding in Knockout Tournaments (Extended Abstract), Thuc Vu,and Yoav Shoham, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1579-1580

round $(r+1)$, i.e., $Q_S^r(i) = \prod_{k=1}^{r} q_S^k(i)$. We consider three different objective functions:

1. *MaxP*– Maximizing the predictive power (i.e., the winning probability of player 1): $\max_{S \in \mathbb{S}} Q_S^{\log n}(1)$
2. *MaxE*– Maximizing the expected value of the winner:
$$\max_{S \in \mathbb{S}} \sum_{i=1}^{n} Q_S^{\log n}(i) \times v_i$$
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:
$$\max_{S \in \mathbb{S}} \sum_{r=1}^{\log n} \sum_{m=(i,j) \in M^r} Q_S^{r-1}(i) \times Q_S^{r-1}(j) \times Rev(m, r)$$
where $M^r$ is the set of all possible matches that can happen in round $r$, and $Rev(m, r)$ is the revenue made by having the match $m$ in round $r$.

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:

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

## 2.3 Solution Types

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 especially focus on two ordinal seedings: $S_1^n = [1, n, (n-1), (n-2), ..., 2]$, and $S_2^n = [...i, (n-i+1), (\frac{n}{2}-i+1), (\frac{n}{2}+i), ...]$. For example, $S_1^8 = [1, 8, 7, 6, 5, 4, 3, 2]$, and $S_2^8 = [1, 8, 4, 5, 2, 7, 3, 6]$.

The seedings decided based on winning probabilities and values of the players are called cardinal solutions. We provide a simple yet effective heuristic algorithm for finding cardinal solutions, and compare these two types of solution.

# 3. EXPERIMENT SETUP

For each of the test cases, we generate the values of the players and the winning probabilities between them so that the monotonicity condition is satisfied. When $n = 8$, there are only 315 different non-duplicate seedings. With this number, we can easily find the optimal solution for any objective function through exhaustive search and use it to evaluate our ordinal and cardinal solutions. However, for $n \geq 16$, this is not possible since there are simply too many seedings, e.g., for $n = 16$, there are $638 \times 10^6$ non-duplicate seedings. In order to evaluate our solutions, we calculate the upper bound for each of the objectives and then compare our solutions to these bounds.

# 4. CARDINAL SOLUTIONS

Our heuristic algorithm is based on the Hill-Climbing approach. The algorithm attempts to improve a given seeding by swapping every pairs of sub-tournament trees of height $k$ ($\forall k = \{1, ..., \log n\}$) to improve the objective value. For each $k$, if there is a new seeding found with a better objective value, the whole process will be repeated for the current $k$ value. Otherwise the seeding with the best objective value so far will be used for the next $k$ value.

# 5. RESULTS

Here we give a list of results. Please see the full version of the paper for more details.

## 5.1 Results for $n = 8$

When $n = 8$, we run $1M$ test cases for each of the objectives. For MaxP, $S_1^8$ seems to be the best candidate. It is optimal in 99.78% of the cases. We also prove that the difference between the values of the seeding $S_1^8$ and the optimal values is at most $\frac{1}{8}$ and this is also the best worst-case difference for all other ordinal seedings. For MaxE, surprisingly, the sequence [1 8 6 7 2 5 3 4] instead of $S_2^8$ has the best chance of being optimal (40.07% of the cases). For the MaxR objective, $S_2^8$ is the best seeding with 22.60%.

Our cardinal solution achieves 100%, 94.99%, and 95.79% optimal for MaxP, MaxE, and MaxR respectively. 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. This shows that our algorithm is efficient and robust.

## 5.2 Results for $n \geq 16$

For each of the objective functions, we provide a provably correct upper bound of the optimal value. We then compare our cardinal and ordinal solutions to this upper bound.

In Figure 1 we show a graph plotting the experimental results for MaxP (the results for MaxE and MaxR are also similar). 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.

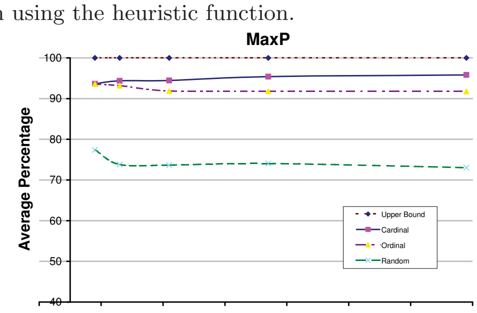


**Figure 1: The average % of objective values of different solutions vs. upper bound for MaxP over 100k tests**