Extending Randomized Single Elimination Bracket to Multiple Prize Vectors

Victoria Graf, Ryan McDowell, Henrique Schechter Vera

Preliminaries

Tournament: directed graph over *n* teams showing outcomes of matches

Tournament ranking rule: ranks the teams in a tournament (possibly randomly)

Prize vector: monotonic vector in [0,1]^n, awards prizes based on ranking

2 kinds of results: **fairness and manipulability**

Preliminaries

Strongly Non-Manipulable (*k*-SNM-α): A tournament rule *r* is *k*-SNM-α if any set of *k* teams cannot increase their expected collective prize winnings by more than α

For Condorcet Consistent rules (an undefeated team wins with probability 1), no rule is 2-SNM-α for $\alpha < 1/3$ [AK 2010]

Preliminaries

Randomized Single Elimination Bracket (RSEB): Select a uniformly random bracket, eliminate losers at each step

Source: Battlebots, https://www.pinterest.com/pin/battle-bots-hobbies-to-try-battle--283656476507369402/

Prior Work

[DFRSW 2022]: Nested Randomized King of the Hill is 2-SNM-⅓ on arbitrary prize vectors

- QuickSort-like tournament rule (not all teams play same number of games)
- Introduced analysis of prize vectors with more than one winner

[SSW 2016]: Randomized Single Elimination Bracket (RSEB) is 2-SNM-⅓ (on tournament with one winner)

Our work: extend RSEB to arbitrary prize vectors through RRB and RCB

Randomized Recursive Bracket (RRB)

On *n* teams, produce a uniformly random perfect matching

Let $T|_W$ be the induced subgraph of T on winning teams. Recur on $T|_W$ with prize vector **p_W** = [p_1, ..., p_{n/2}]. Separately, recur on losing teams with the vector **p_L** = [p_{n/2+1}, ..., p_n].

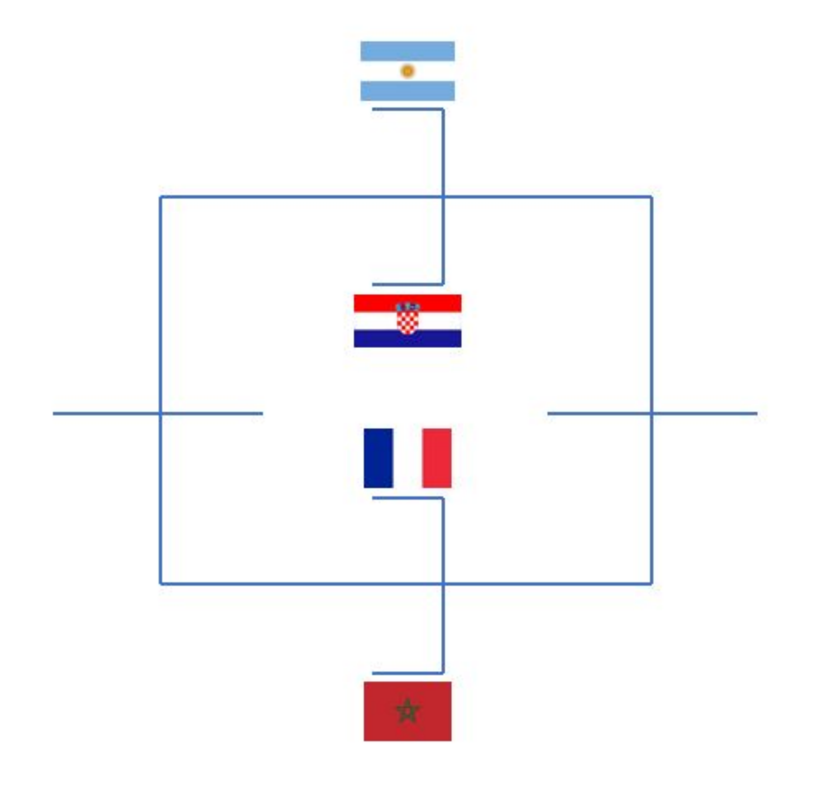
If *n* is not a power of 2, create dummy teams as necessary who all lose to the original *n* teams, and match all of these dummy teams to actual teams

Randomized Complete Bracket (RCB)

Select a random ordering of teams $t_1, t_2, ..., t_n$

Given two pairs a_1 beats b_1 and a_2 beats b_2 at the kth round of the bracket, then a_1 plays a_2 in the $k+1$th round, b_1 plays b_2 in the $k+1$th round

Bracket is completely defined by the initial ordering of teams

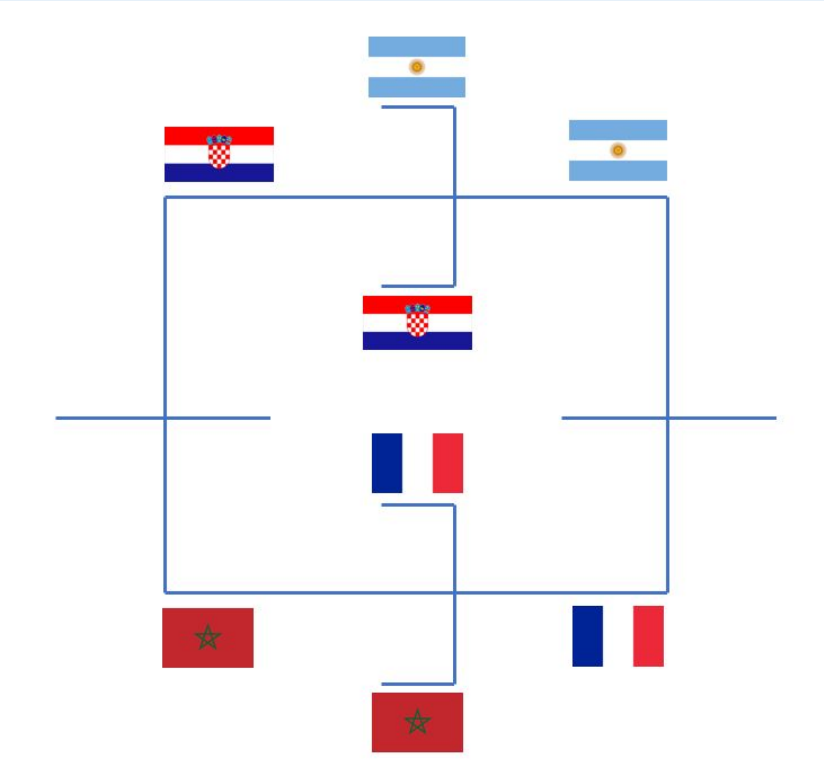


Randomized Complete Bracket (RCB)

Select a random ordering of teams $t_1, t_2, ..., t_n$

Given two pairs a_1 beats b_1 and a_2 beats b_2 at the kth round of the bracket, then a_1 plays a_2 in the $k+1$th round, b_1 plays b_2 in the $k+1$th round

Bracket is completely defined by the initial ordering of teams

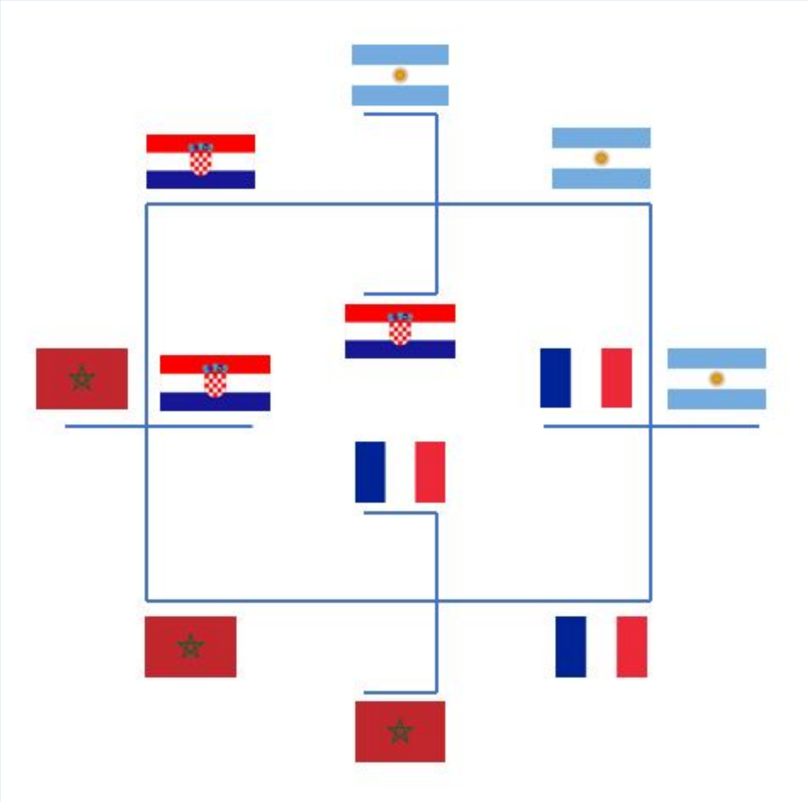


Randomized Complete Bracket (RCB)

Select a random ordering of teams $t_1, t_2, ..., t_n$

Given two pairs a_1 beats b_1 and a_2 beats b_2 at the kth round of the bracket, then a_1 plays a_2 in the $k+1$th round, b_1 plays b_2 in the $k+1$th round

Bracket is completely defined by the initial ordering of teams



RCB is 2-SNM-½

Lemma: teams *i* and *j* cannot manipulate the tournament if they start on opposite sides of an RCB bracket (they play each other only in the last round)

Lemma: if *i* and *j* manipulate the tournament, they gain at most 1 collectively

(follows from monotonicity)

Theorem: RCB is 2-SNM-½

> There are $n!$ initial brackets for RCB, chosen from uniformly
>
> $\frac{1}{2} n^2(n-2)! > \frac{1}{2} n!$ initial brackets where *i* and *j* start on opposite sides
>
> Expected gain = (probability they can collude) * (gain if collude) < ½

RCB and RRB are 2-SNM-⅓ on BPoT prize vectors

Binary Power-of-Two (BPoT) Prize Vectors: A prize vector with $n=2^m$ total entries and 2^k entries s.t. $p_i=1$ for some integer $0 \leq k \leq m$, $p_i=0$ *o.w.*

Lemma: Under a BPoT vector with 2^k entries and 2^i ones in a RCB or RRB tournament, a team gets a prize of 1 if and only if they win their first $k - i$ matches

Can split the bracket into 2^i groups, RRB and RCB is then equivalent to RSEB (since there is one prize for each group), which is 2-SNM-⅓

RCB / RRB are manipulable on the Borda vector

Borda Prize Vector: prize vector with *n* entries where $p_i = (n-i)/(n-1)$

[DFRSW 2022]: NRKotH is non-manipulable by any set of teams on the Borda vector

Counterexample for RCB / RRB - teams A, B, C, D: A beats B, C; B beats C, D; C beats D; D beats A. B and D can collude

Expected return without colluding: $\frac{1}{3}(0 + \frac{1}{3}) + \frac{1}{3}(0 + \frac{2}{3}) + \frac{1}{3}(1 + \frac{2}{3}) = 8/9$

Expected when colluding: $\frac{1}{3}(0 + \frac{1}{3}) + \frac{1}{3}(1 + \frac{1}{3}) + \frac{1}{3}(1 + \frac{2}{3}) = 11/9$

Towards 2-SNM-⅓

Recall: 2-SNM-⅓ is optimal for Condorcet consistent tournament rules

Were unable to prove for RCB, RRB (progress in paper on proof by injective mappings, following [SSW ‘16])

Seems likely based on simulation results (no counterexamples found after running a simulation for 24 hours)

Fairness: RCB and RRB are not cover-consistent

Definition: Team *i* **covers** team *j* in tournament *T* if *i* beats *j* and every team that *j* beats

Definition: A tournament ranking rule is **cover-consistent** if for all *T*, and all *i, j* such that *i* covers *j* in *T*, *i* is ahead of *j* in the ranking with probability 1

Counterexample: Suppose A beats all; B beats C and D; C beats D. Note that B covers C

If initial bracket is A-B, C-D (probability $\frac{1}{3}$), then C will be ranked ahead of B

RCB and RRB are not consistent under expectation

Consistent Under Expectation: if team *u* beats *k* teams in *T*, *u* is expected to rank above exactly *k* teams

Counterexample: A beats C, D; B beats A; D beats C

Possibilities of first round pairings (each happens with ⅓ probability):

- A - B: A gets 3rd
- A - C: A gets 1st
- A - D: A gets 1st

Expected ranking: $5/3 \neq 2$

Conclusion

We extended RSEB to tournaments with multiple winners via RCB and RRB

Unlike NRKotH, teams play an equal number of games in RCB, RRB; though ranking is intuitively less fair than NRKotH in other ways (cover-consistency)

Future: exploring whether RCB and RRB are minimally manipulable (2-SNM-⅓)

Questions?