styopa@stanford.edu

Query Complexity of Approximating Tournament Winner

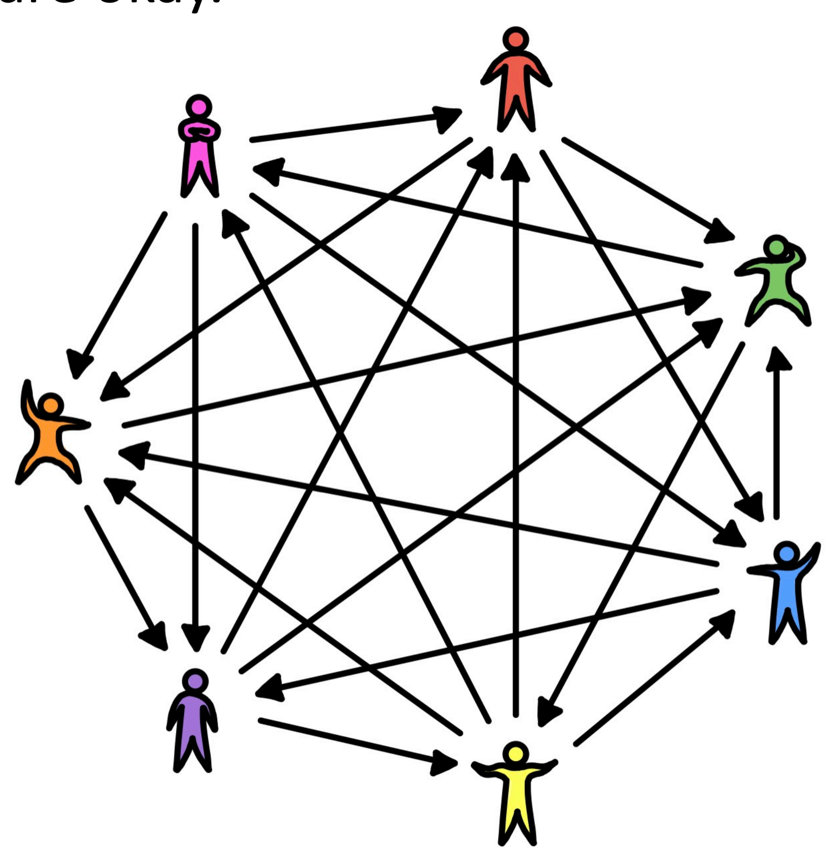
Styopa Zharkov, Bharath Namboothiry, Daniel Rebelsky, Li-Yang Tan, Moses Charikar

Stanford University Department of Computer Science

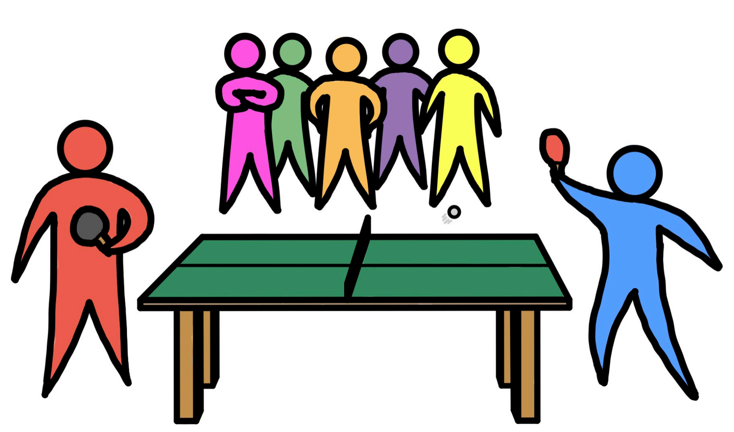
The Question:

How many games need to be played to select an approximate winner in a tournament?

A *tournament* is a complete directed graph, where the vertices denote players and the edges denote who beats who. Note that cycles are okay.



Imagine that our players are playing eternal ping-pong where the winner stays and the loser gets replaced by a random player. We can define each player's score to be the proportion of games that they win. These are called *Markov scores.* The player with the highest score is the *Markov winner.*



How many of the tournament's edges do we need to know (i.e. query) to select a player whose Markov score isn't too small in comparison to the Markov winner's score?

Main Result:

For a constant ratio, all games up to a constant factor must be played.

<u>*Theorem 1*</u>*:* In an n-player tournament, the query complexity of finding a player with r times the maximum Markov score is $\Omega(rn^2)$.

Note that

- a tournament has $O(n^2)$ edges total, so if r is constant as n grows, this bound is tight.
- An $\Omega(n)$ bound is easy to show, so the best known bound is actually $\Omega(\max(n, rn^2))$.

Previous Work:

Another common measure of player strength is the *Copeland score*, which is just the player's out-degree. Our work fills the space left by the following two results:

- Query complexity of *finding* the Markov or Copeland winner is $\Omega(n^2)$ (Dey 2016)
- Query complexity of *r-approximating* the Copeland winner is $\Omega((rn)^2)$ (Hulett 2019)

	Copeland	Markov
finding	$\Theta(n^2)$	$\Theta(n^2)$
r-approx.	$\Theta(\max(1, rn)^2)$	$\Omega(\max(n, rn^2))$

our result

Another look at Markov Scores:

We can equivalently define Markov scores through a random process on the tournament. The Markov transition matrix is

$$Q = (G + Co)/(n-1),$$

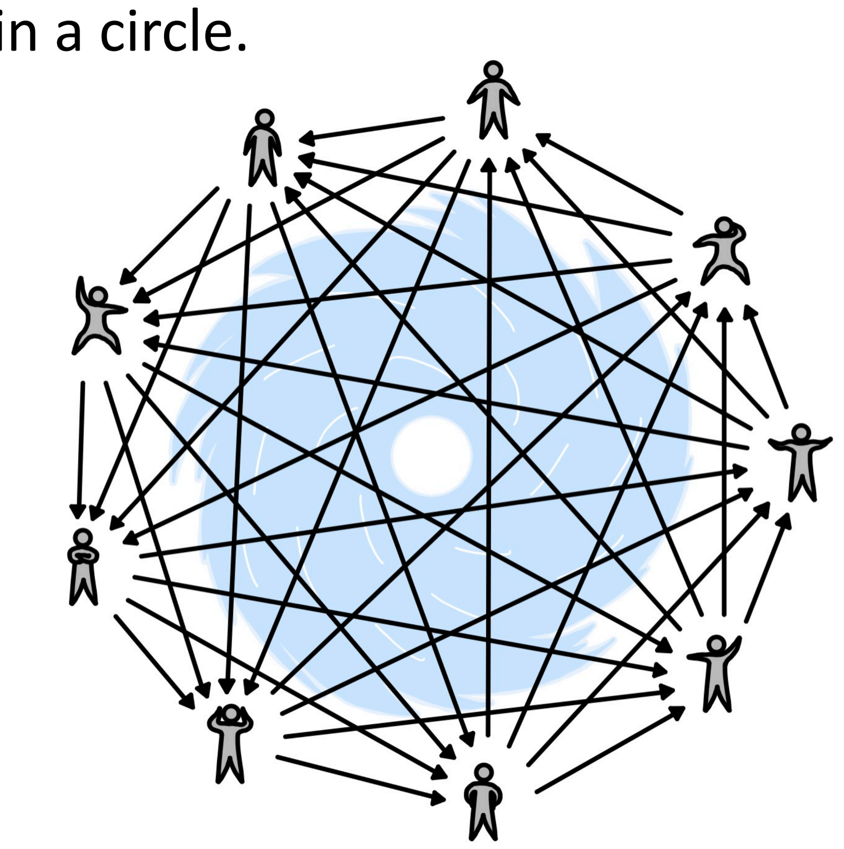
where G is the graph's adjacency matrix and Co is a diagonal matrix where $Co_{i,i}$ is player i's Copeland score. Then the unique distribution p, where

$$Qp = p,$$

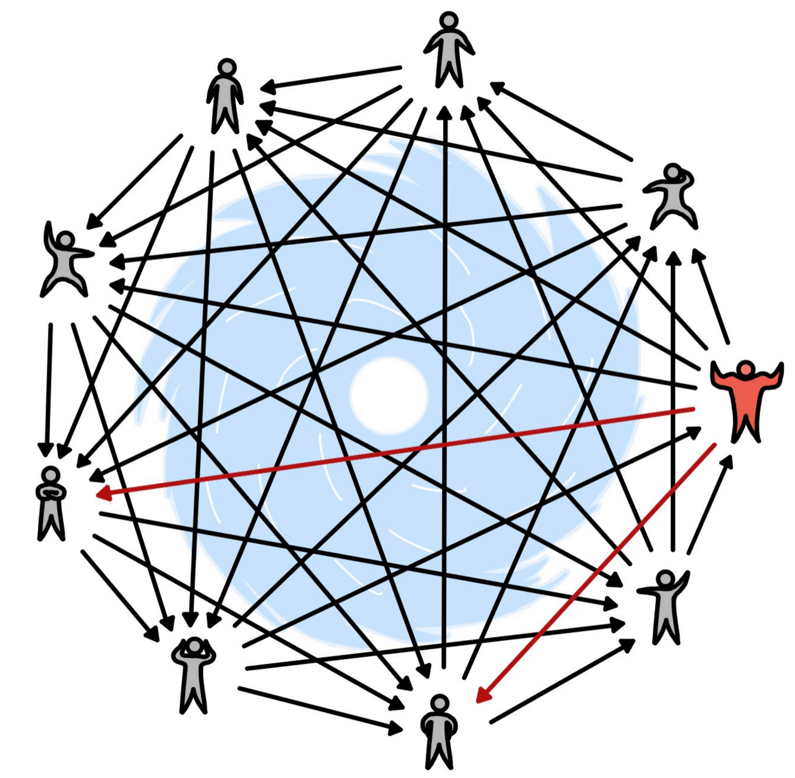
is the stationary distribution. Player i's score is p_i.

Proof Idea:

A *cyclone* is a tournament where each player beats the next $(n-1)/2$ players and loses to the previous $(n-1)/2$ in a circle.



A *δ-flipped cyclone* is constructed by taking a cyclone, selecting a vertex, and flipping any $\lfloor \delta n \rfloor$ of its in-edges. The selected vertex is called the *strong player* of this tournament.



The Idea: How does a potential algorithm preform on δ-flipped cyclones?

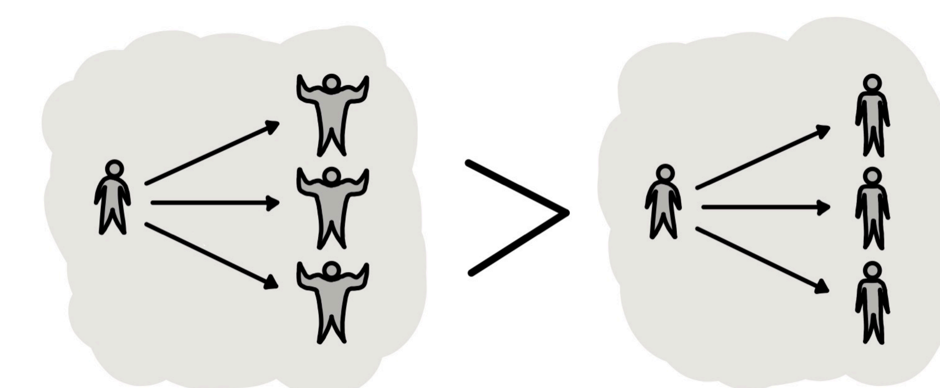
Theorem 1 follows from the following lemmas:

1. With the right choice of δ, the ratio between the Markov score of any non-strong player and that of the strong player is less than r.
2. Any algorithm can't determine the strong player without querying $\Omega(rn^2)$ edges.

Why this Model?

There are many ways to define a tournament and many ways to measure a player's strength. Why is our model relevant?

- Simplest definition of a tournament.
- Markov scores are resistant to small changes in the graph
- Markov scores distinguish between beating weak players and beating strong players (unlike Copeland scores)



Ongoing and Future Work:

- The current proof of the first lemma uses an ugly analysis of Q's spectral gap, so we are working on a nicer proof through Fourier decomposition.
- Is our bound tight for non-constant r?
- What if we allow randomized algorithms?
- What if we look at the average over some distribution of tournaments instead of worst-case?
- Can this result be applied in a statement about general Markov processes?

Acknowledgements:

A special thanks to my advisors Li-Yang Tan and Moses Charikar for their support throughout the project, Victor Kleptsin for some helpful discussions, and the CURIS program for this opportunity.