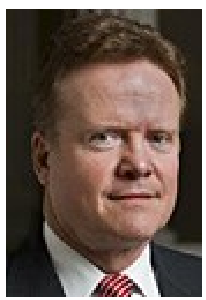
Voronoi Choice Games

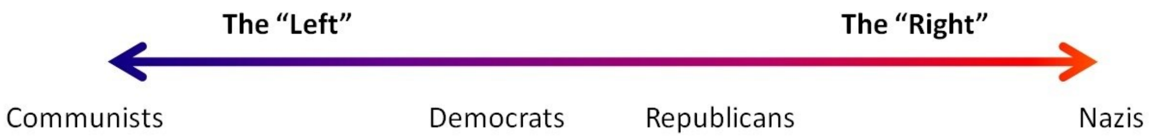
Meena Boppana, Rani Hod, Michael Mitzenmacher, Tom Morgan

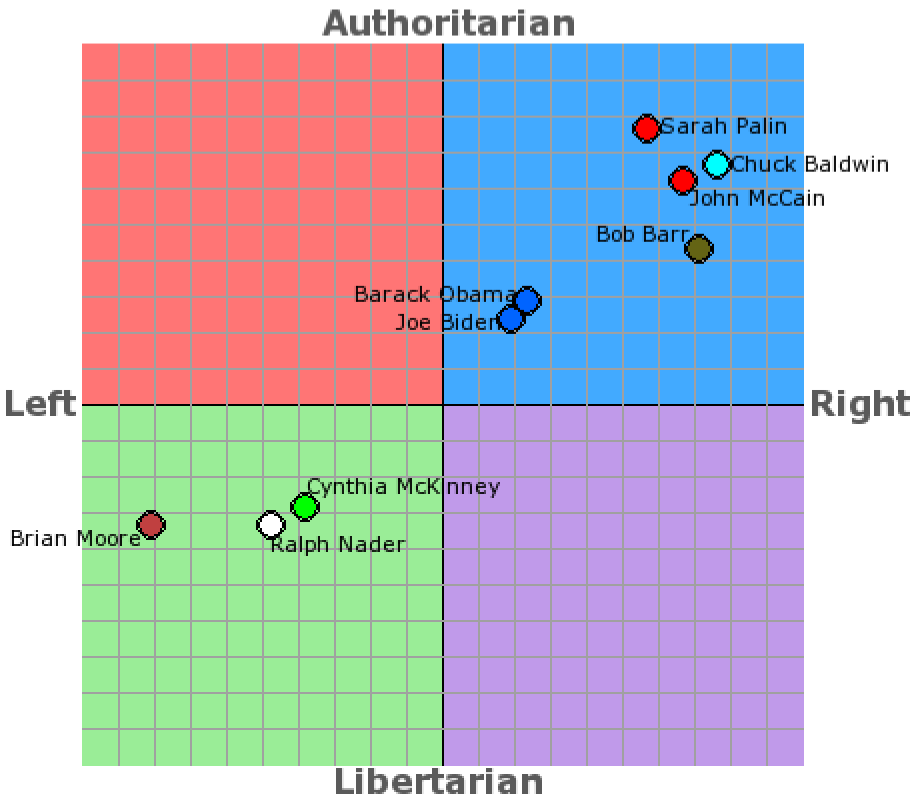
7/12/16

Political Candidate Selection



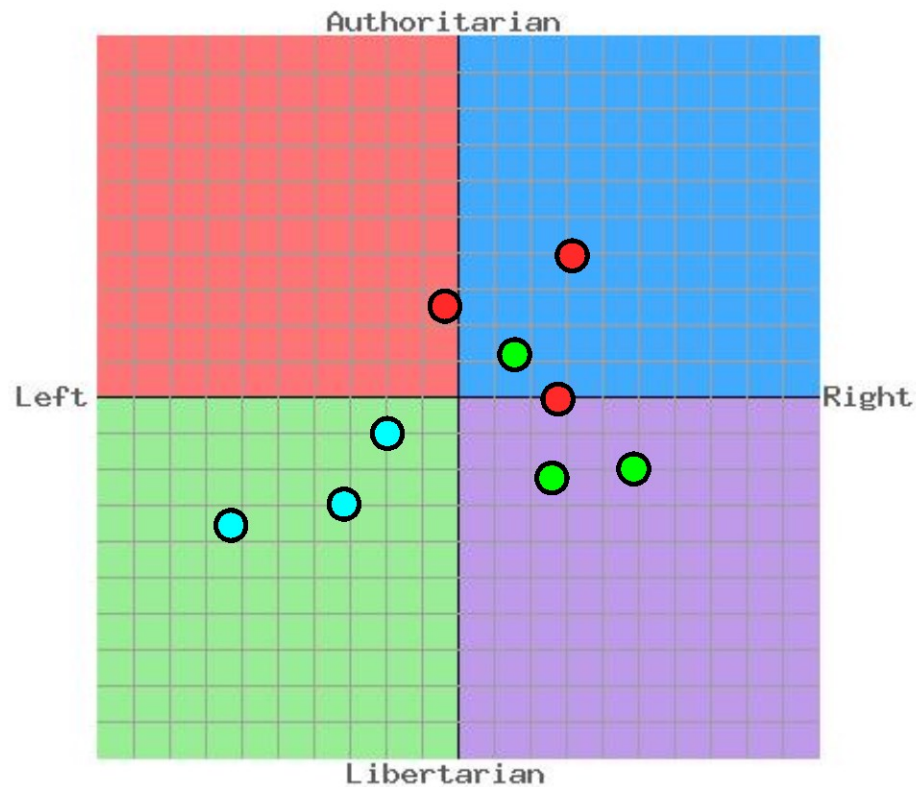
Political Positions





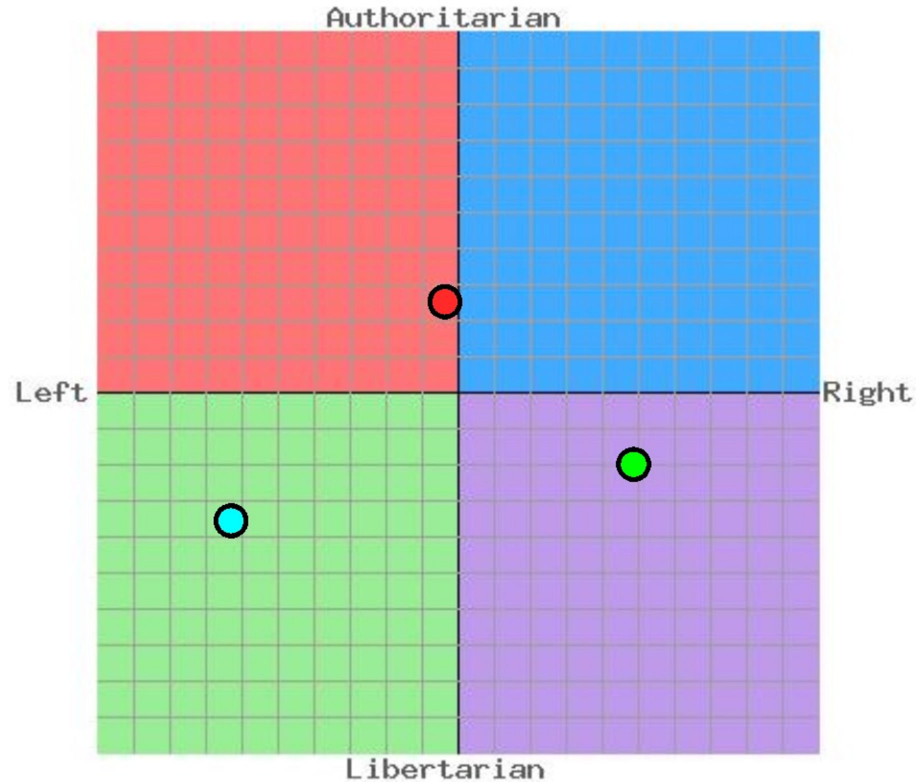
Candidates Selection Game

- Political candidate selection
  - **n** parties, each with a separate pool of candidates
  - Simultaneously pick candidates so as to maximize votes



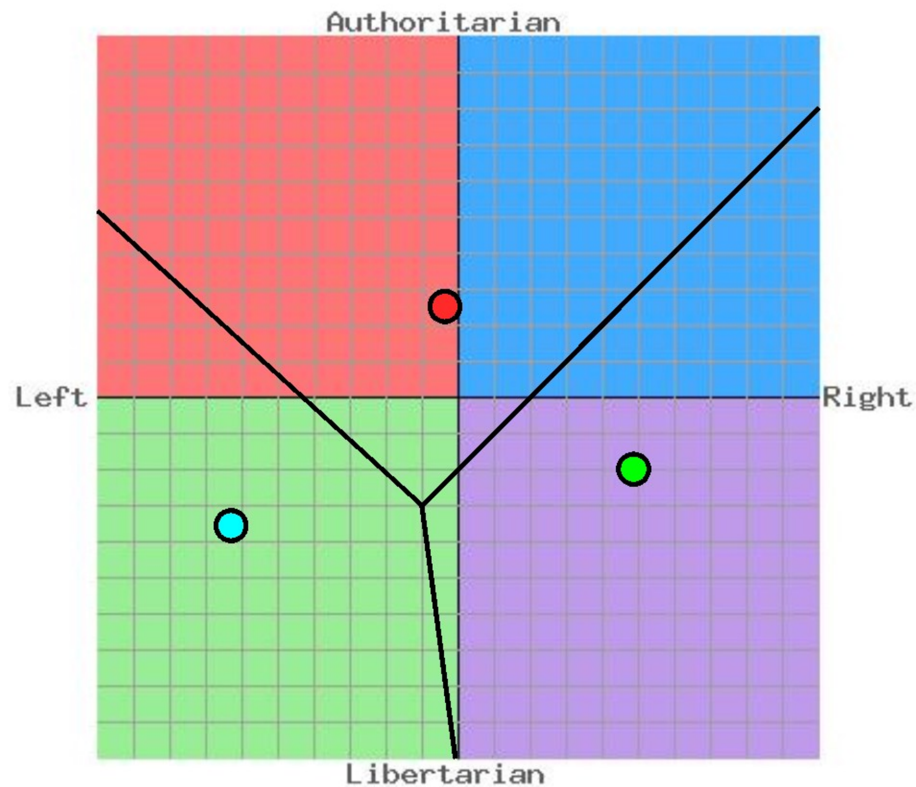
Candidates Selection Game

- Political candidate selection
  - **n** parties, each with a separate pool of candidates
  - Simultaneously pick candidates so as to maximize votes



Candidates Selection Game

- Political candidate selection
  - **n** parties, each with a separate pool of candidates
  - Simultaneously pick candidates so as to maximize votes



Models

- k-D Voronoi Choice Game
  - **k** dimensional unit torus
  - **n** players
  - Each player has **m** (small constant) points
  - Each player simultaneously picks a point
  - Utility = area of point's cell in Voronoi diagram
- One Way 1-D Voronoi Choice Game
  - Utility = distance to next point clockwise

Related Games

- Voronoi Games
  - Two player
  - Unconstrained choices
  - Sequential games
- Hotelling Games
  - AKA competitive location games
  - Sometimes limited choices, but always symmetric
  - On line or graph
- Main difference with our model: **player asymmetry**

Summary of Results

- Proof of existence of pure Nash equilibria for the 1-D Voronoi Choice Game
- NP-hardness of pure Nash equilibria in One Way 1-D Voronoi Choice Game
- Strong bounds on the expected number of pure Nash equilibria in random versions of the One Way 1-D Voronoi Choice Game
- Polytime algorithm for finding correlated equilibria in all 1-3 dimensional versions of the game

Summary of Results

- Proof of existence of pure Nash equilibria for the 1-D Voronoi Choice Game
- NP-hardness of pure Nash equilibria in One Way 1-D Voronoi Choice Game
- **Strong bounds on the expected number of pure Nash equilibria in random versions of the One Way 1-D Voronoi Choice Game**
- **Polytime algorithm for finding correlated equilibria in all 1-3 dimensional versions of the game**

Random Pure Nash Equilibria

- Model: One Way 1-D (n players, m choices each)
- Each player's points are **independently** chosen **uniformly** at random
- Motivation: "power of choice" in load balancing
- The expected number of PNE in the One Way 1-D Voronoi Choice Game is in $[.19^{m-1} m, m]$
  - Small constant (for small m), no dependence on n

Random Pure Nash Equilibria

A player's point **choice** is **stable** in a configuration if they won't deviate from it

A circle **arc** is **stable** if the point that starts it is stable

E[# PNE] = m^n Pr[first choices are stable]

= m^n E[Pr[first choices are stable|positions of first choices]]

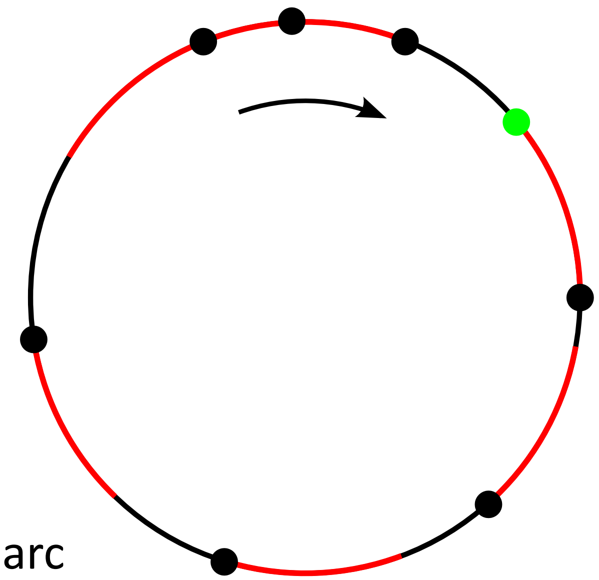
= m^n E[$\prod_{i=1}^{n}$ Pr[ith player is stable|positions of first choices]]

= m^n E[$\prod_{i=1}^{n}$ Pr[ith smallest arc is stable|positions of first choices]]

Random Pure Nash Equilibria

Pr[ith smallest arc is stable|positions of first choices]?

Red are regions which green player would not deviate to

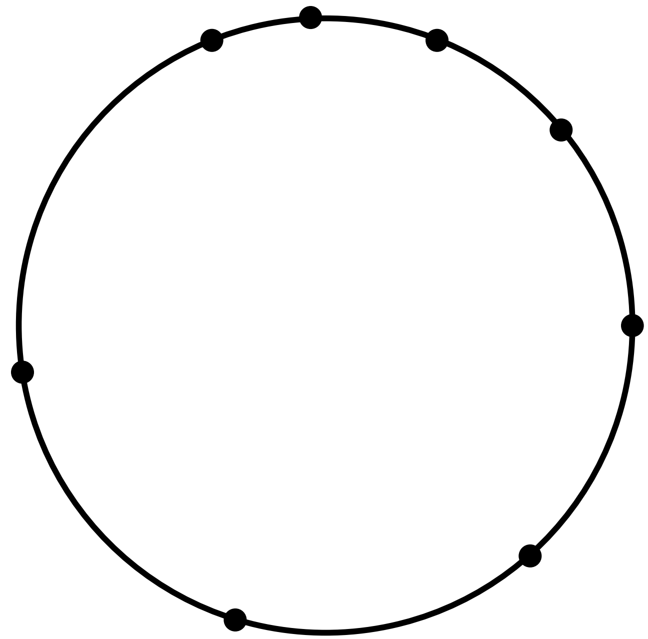


A_i = length of the ith smallest arc

Pr[ith smallest arc is stable|positions of first choices]

= $\left((n-i)A_i + \sum_{j=1}^{i} A_j - \min(A_i, \text{length of arc before i})\right)^{m-1}$

Random Pure Nash Equilibria



- Now a question about the lengths of **random arcs on circles**
- Distance between points is **exponentially distributed**
- Take order statistics of those exponentials
- Length of ith smallest arc $A_i \sim \frac{1}{S_n}\left(\frac{X_n}{n} + \frac{X_{n-1}}{n-1} + \cdots + \frac{X_{n-i+1}}{n-i+1}\right)$
- X_j are i.i.d. exponential random vars with mean 1
- $S_j = X_1 + \cdots + X_j$

Random Pure Nash Equilibria

- Pr[ith smallest arc is stable|positions of first choices]

  $$= \left((n-i)A_i + \sum_{j=1}^{i} A_j - \text{min}(A_i, \text{length of arc before i}) \right)^{m-1}$$

  $$\in \left[\left(\frac{S_i}{S_n} - A_i \right)^{m-1}, \left(\frac{S_i}{S_n} \right)^{m-1} \right]$$

- E[# PNE] ≤ m^n E$\left[\prod_{i=1}^{n} \left(\frac{S_i}{S_n} \right)^{m-1} \right]$ = m
- $S_n, \left(\frac{S_1}{S_2} \right), \left(\frac{S_2}{S_3} \right), \ldots, \left(\frac{S_{n-1}}{S_n} \right)$ are independent

- E[# PNE] ≥ m^n E$\left[\prod_{i=1}^{n} \left(\frac{S_i}{S_n} - A_i \right)^{m-1} \right]$

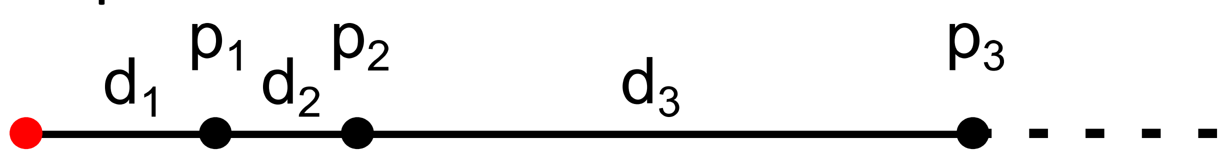
  ≥ m^n E$\left[\prod_{i=1}^{n} \left(\frac{c_i S_i}{S_n} \right)^{m-1} \right]$ ≥ $.19^{m-1}$ m

Correlated Equilibria

- A **correlated equilibrium** (CE) is a generalization of Nash equilibrium to allow player strategies to be correlated
- (Papadimitriou & Roughgarden, 2008) and (Jiang & Leyton-Brown, 2013) developed polytime algorithm for CE in games of **polynomial-type**
- Requires a polytime oracle for computing the **expected utility** given a **product distribution** over strategies (each player is independent)

1D Correlated Equilibria

- **Goal:** compute the expected utility over a product distribution for 1-D Voronoi Choice Games
- For each player point, compute the expected distance from that point to the next chosen point in each direction
- Sort the points and iterate through them in order, computing the probability that each point is the first chosen point

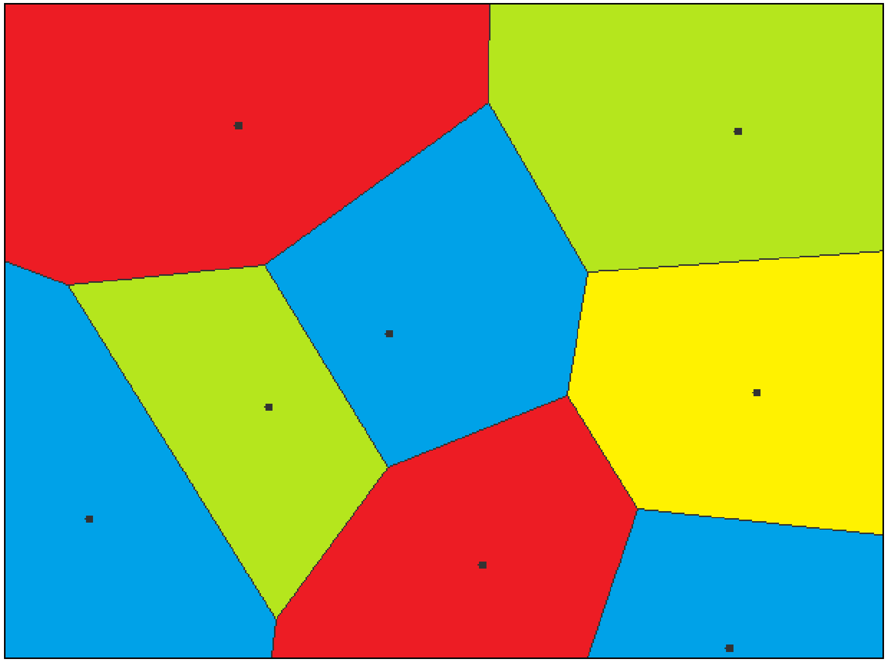


$$E[\text{right dist}] = p_1d_1+(1-p_1)p_2d_2+(1-p_1)(1-p_2)p_3d_3+\ldots$$

- Stop once we reach a player's last point

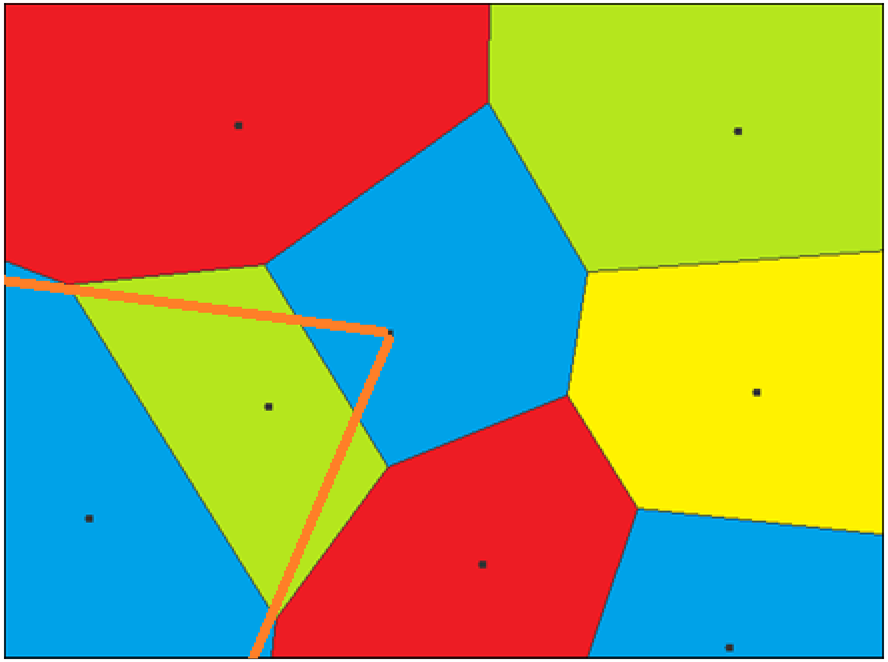
2D Correlated Equilibria

- What about 2D?
- A given Voronoi cell can be bounded by many other players' points



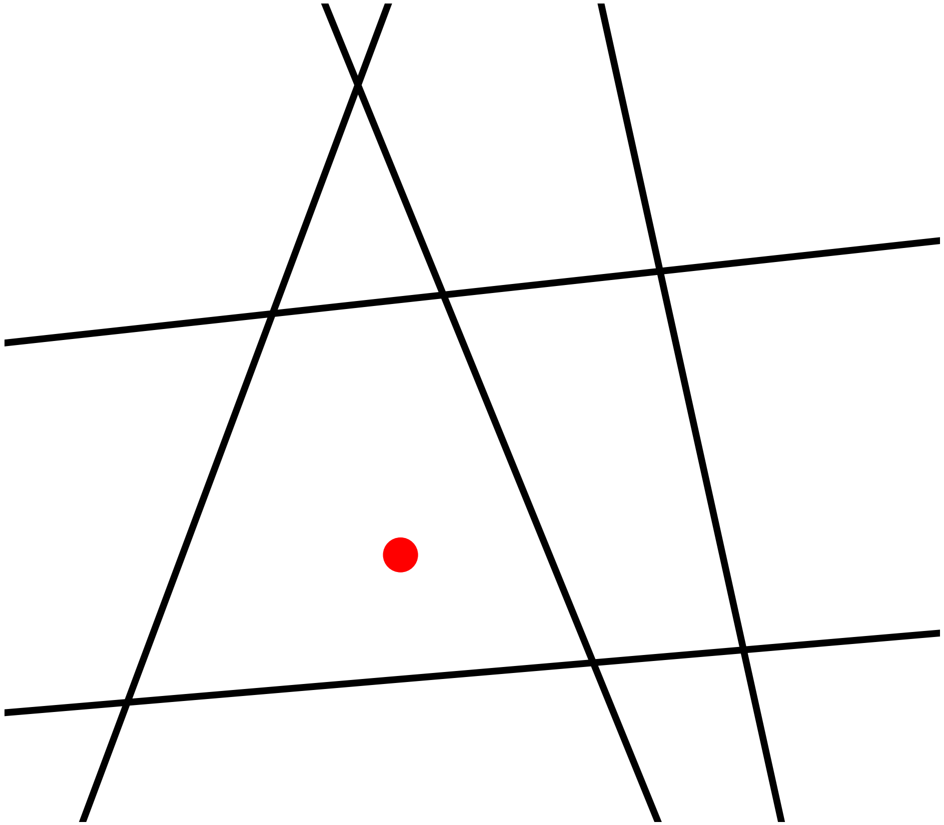
2D Correlated Equilibria

- **Idea**: partition space into regions where the 1D strategy works



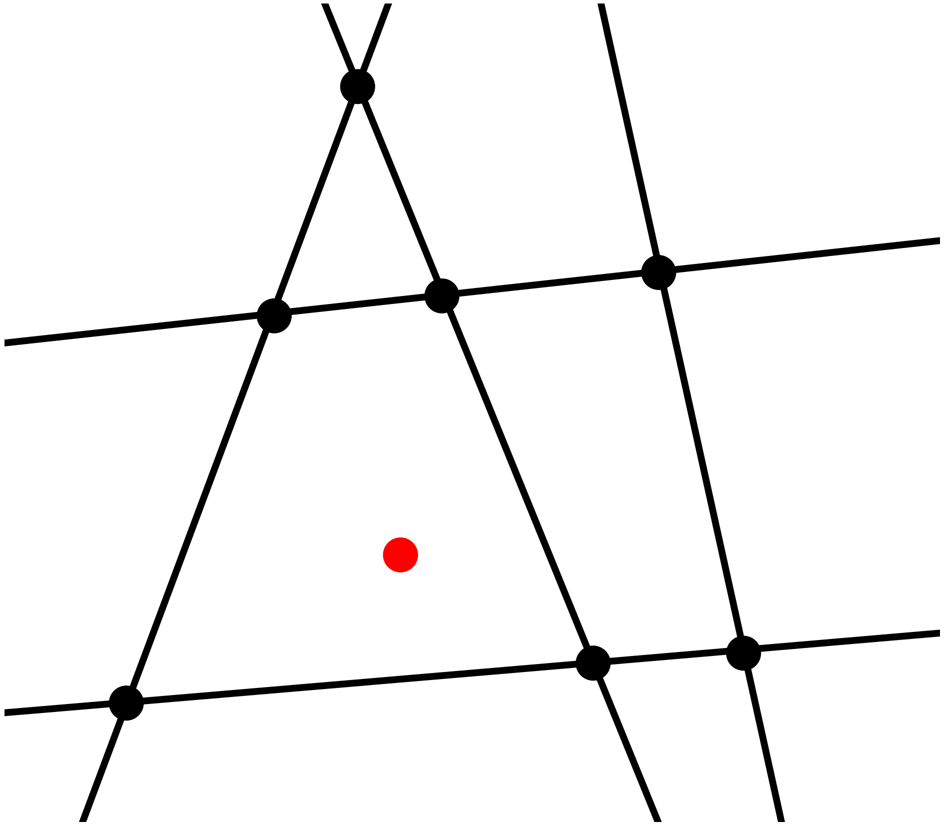
2D Correlated Equilibria

- Compute the expected utility of a given point:
  - Each other point yields a boundary line



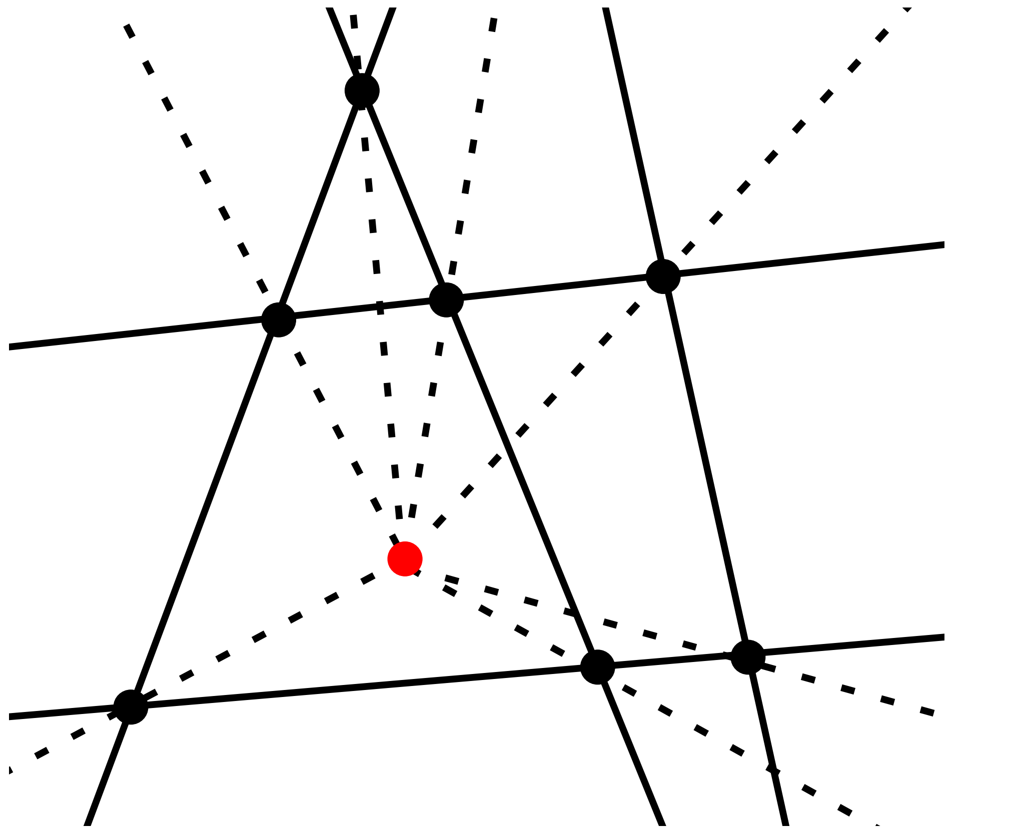
2D Correlated Equilibria

- Compute the expected utility of a given point:
  - Each other point yields a boundary line
  - Compute all intersections between boundary lines



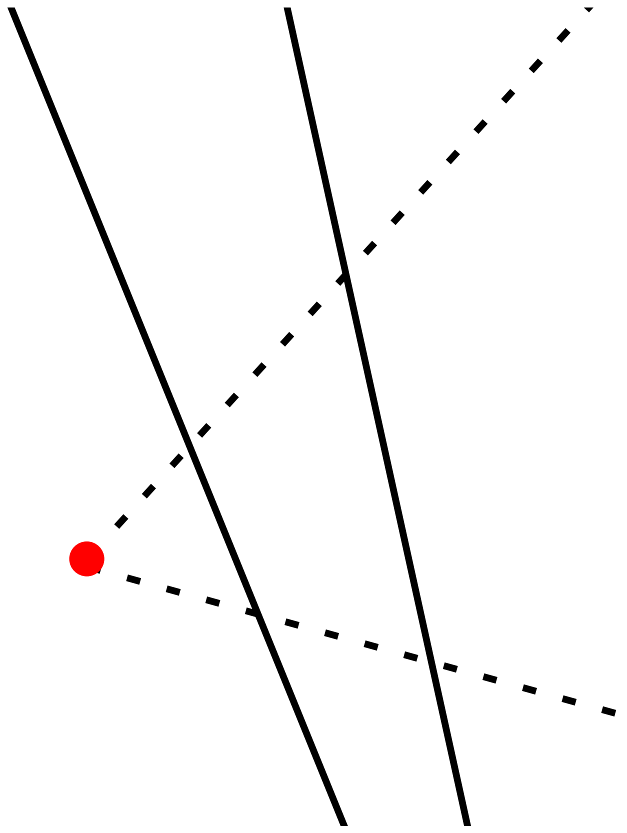
2D Correlated Equilibria

- Compute the expected utility of a given point:
  - Each other point yields a boundary line
  - Compute all intersections between boundary lines
  - Sort those points by angle
  - Decompose area into regions in between angles



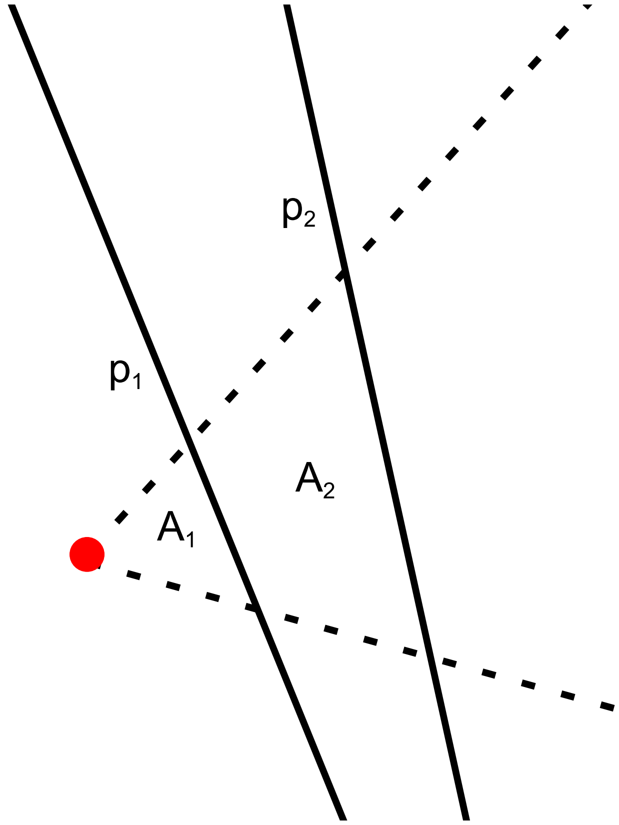
2D Correlated Equilibria

- Compute the expected utility of a given point:
  - Each other point yields a boundary line
  - Compute all intersections between boundary lines
  - Sort those points by angle
  - Decompose area into regions in between angles
  - In each region iterate through boundary lines as in 1D



2D Correlated Equilibria

- Compute the expected utility of a given point:
  - Each other point yields a boundary line
  - Compute all intersections between boundary lines
  - Sort those points by angle
  - Decompose area into regions in between angles
  - In each region iterate through boundary lines as in 1D



$$E[\text{area in region}] = p_1A_1 + (1-p_1)p_2A_2 + \ldots$$

2D Correlated Equilibria

- Time = $O(n^3 \log n)$
  - $O(n)$ boundary lines → $O(n^2)$ regions
  - $O(n \log n)$ time per region to sort lines and iterate
- Can reduce to $O(n^3)$ by not resorting each time
  - Only one swap between adjacent regions
- Same basic principles get us $O(n^5 \log n)$ in 3D

Open Questions

- Probability of any PNE in random model?
- Correlated equilibria in 4D+?
- What about mixed Nash equilibria?
- Sequential versions of these games?

Thank You