15-780: Graduate AI

Lecture 22. Learning in Games

Geoff Gordon (this lecture)
Ziv Bar-Joseph
TAs Michael Benisch, Yang Gu

Recap

- *On Monday, we talked a lot about solution concepts for matrix games*
  - *Support enumeration for Nash equilibria*
  - *LPs for finding correlated equilibria*
  - *Pie-splitting for deciding which of the many equilibria to follow*

Recap

- *We also talked about different learning algorithms based on opponent modeling:*
  - *fictitious play*
  - *fancier versions of FP: e.g., use a classifier to predict opp actions*
  - *rational learning*

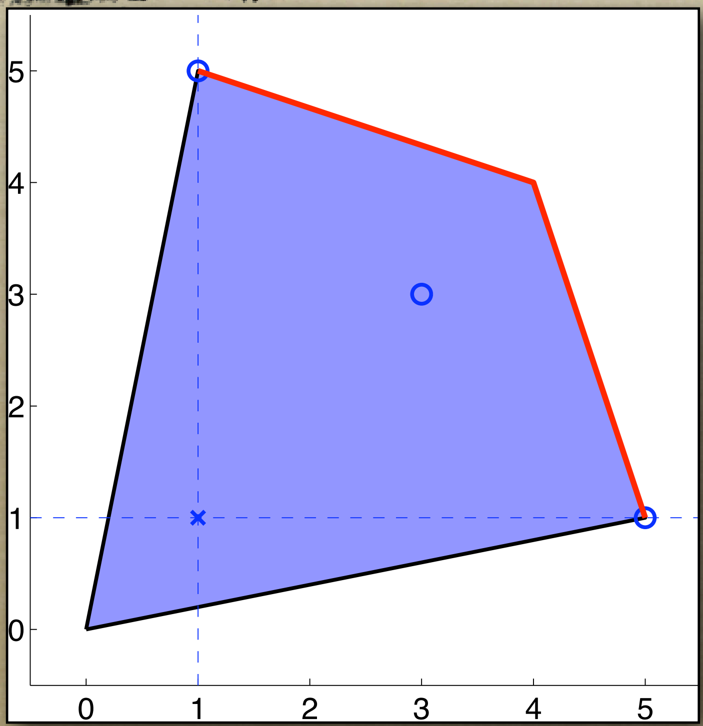
Recap

- *All opp-modeling learners had only weak guarantees: at best, convergence to some unspecified equilibrium in self-play*
  - *not necessarily Pareto*
  - *no guarantees in non-self play (exc RL)*
  - *might not converge—eg., FP on Shapley*
- *And, we had to keep our exact modeling method secret (or risk being exploited)*

This lecture

- *Want to show two things*
  - *First, algorithms with slightly stronger guarantees for matrix games*
  - *Second, since the world isn't a matrix game, applications to more realistic models*

CE ex w/ info hiding necessary



	L	R
T	5,1	0,0
B	4,4	1,5

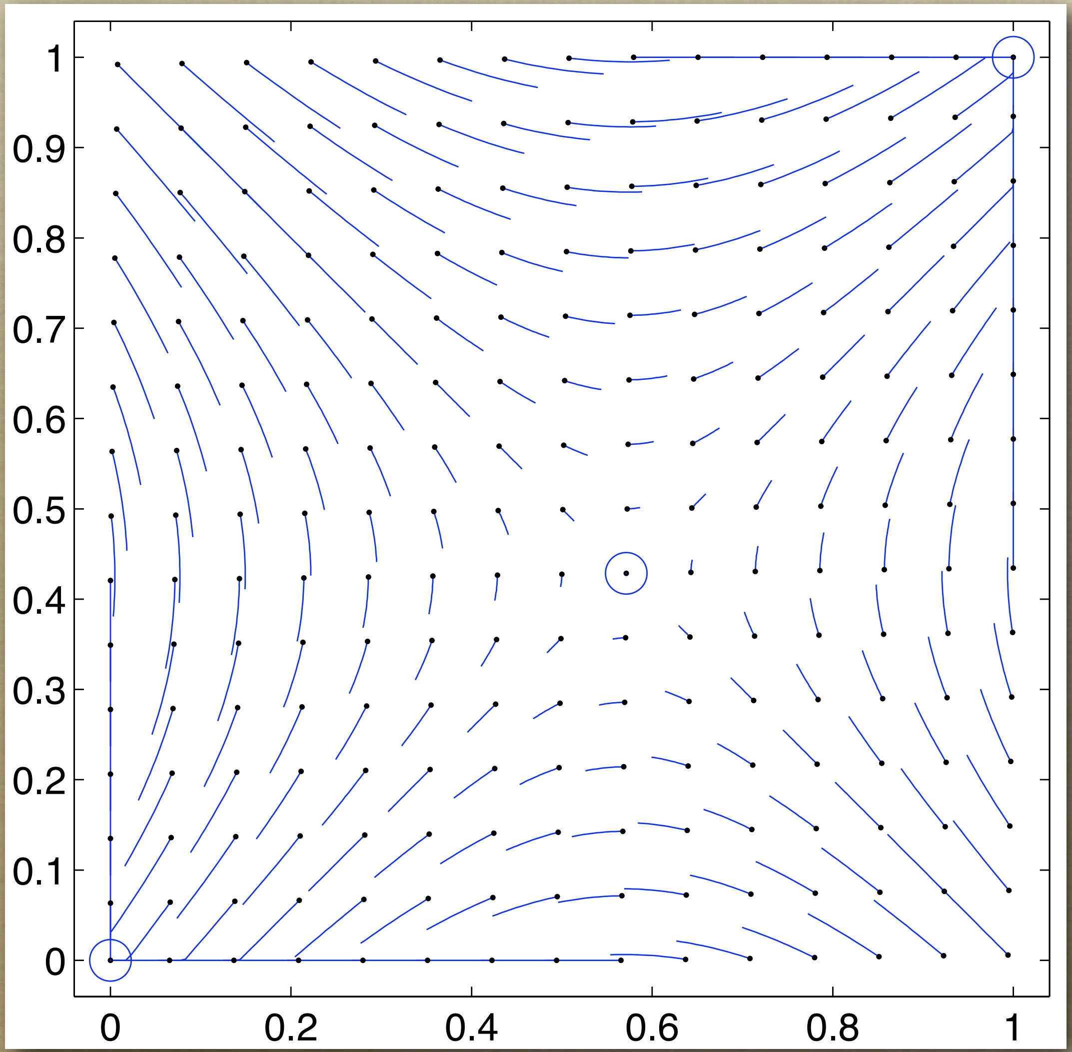
- *3 Nash equilibria (circles)*
- *CEs include point at TR: 1/3 on each of TL, BL, BR (equal chance of 5, 1, 4)*

Policy gradient

Next try

- *What can we do if not model the opponent?*
- *Next try: **policy gradient** algorithms*
- *Keep a parameterized policy, update it to do better against observed play*
- *Note: this seems irrational (why not maximize?)*

Gradient dynamics for Lunch



Theorem

- *In a 2-player 2-action repeated matrix game, two gradient-descent learners will achieve payoffs and play frequencies of some Nash equilibrium (of the stage game) in the limit*

Satinder Singh, Michael Kearns, Yishay Mansour. Nash Convergence of Gradient Dynamics in General-Sum Games. UAI, 2000

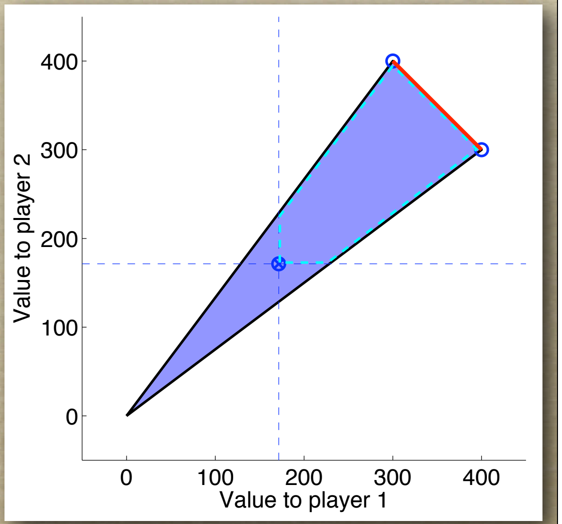
Theorem

- *A gradient descent learner with appropriately-decreasing learning rate, when playing against an* ***arbitrary*** *opponent, will achieve at least its safety value. When playing against a* ***stationary*** *opponent, it will converge to a best response.*

Gordon, 1999; Zinkevich, 2003

Discussion

- *Works against arbitrary opponent*
- *Gradient descent is a member of a class of learners called* ***no-regret algorithms*** *which achieve same guarantee*
- *Safety value still isn't much of a guarantee, but…*



Pareto

- *What if we start our gradient descent learner at (one side of) an equilibrium on the Pareto frontier?*
- *E.g., start at "always Union Grill"*
- *In self-play, we stay on Pareto frontier*
- *And we still have guarantees of safety value and best response*
- *Same idea works for other NR learners*

Pareto

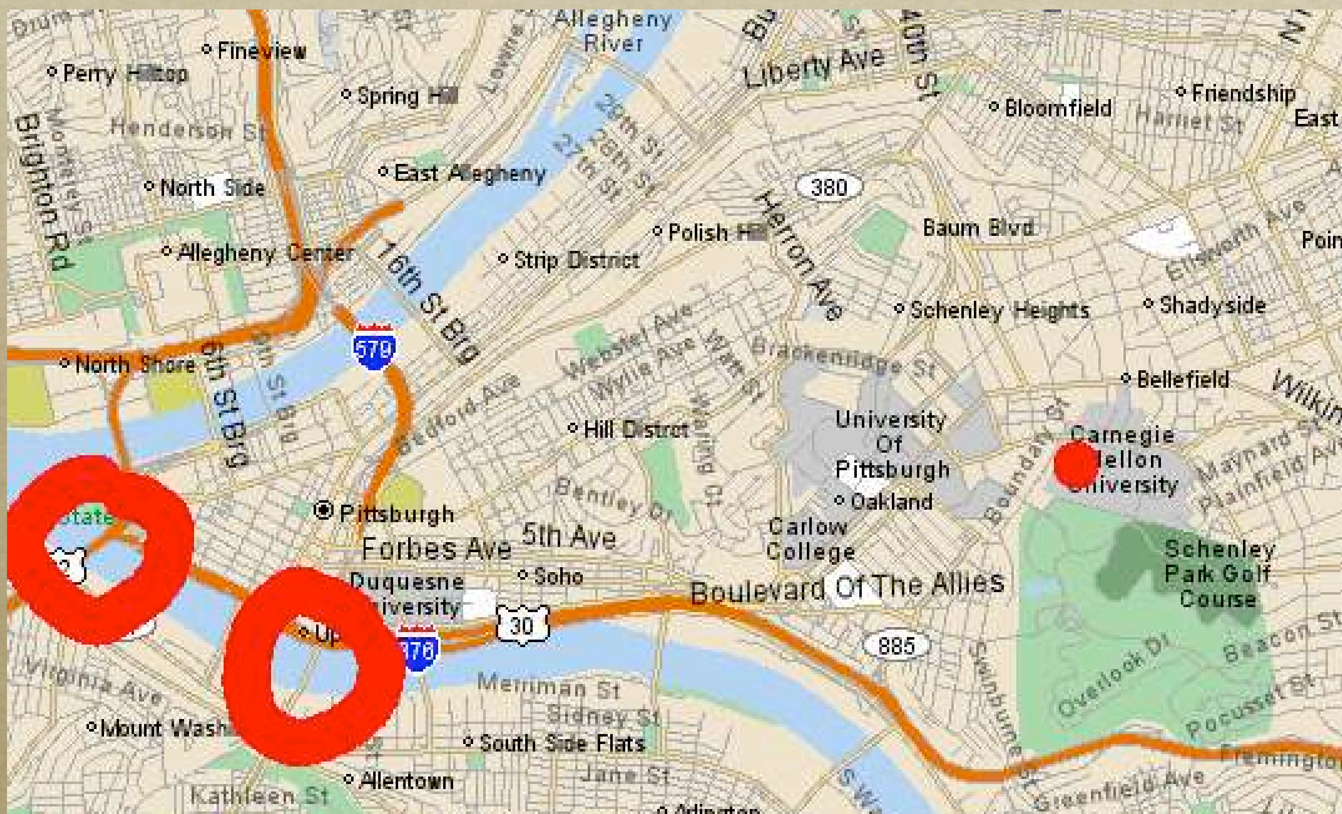
- *First learning algorithm we've discussed that guarantees Pareto in self-play*
- *Only a few algorithms with this property so far, all since about 2003 (Brafman & Tennenholtz, Powers & Shoham, Gordon & Murray)*
- *Can't really claim it's "negotiating" — would like to be able to guarantee something about accepting ideas from others*

Open problems

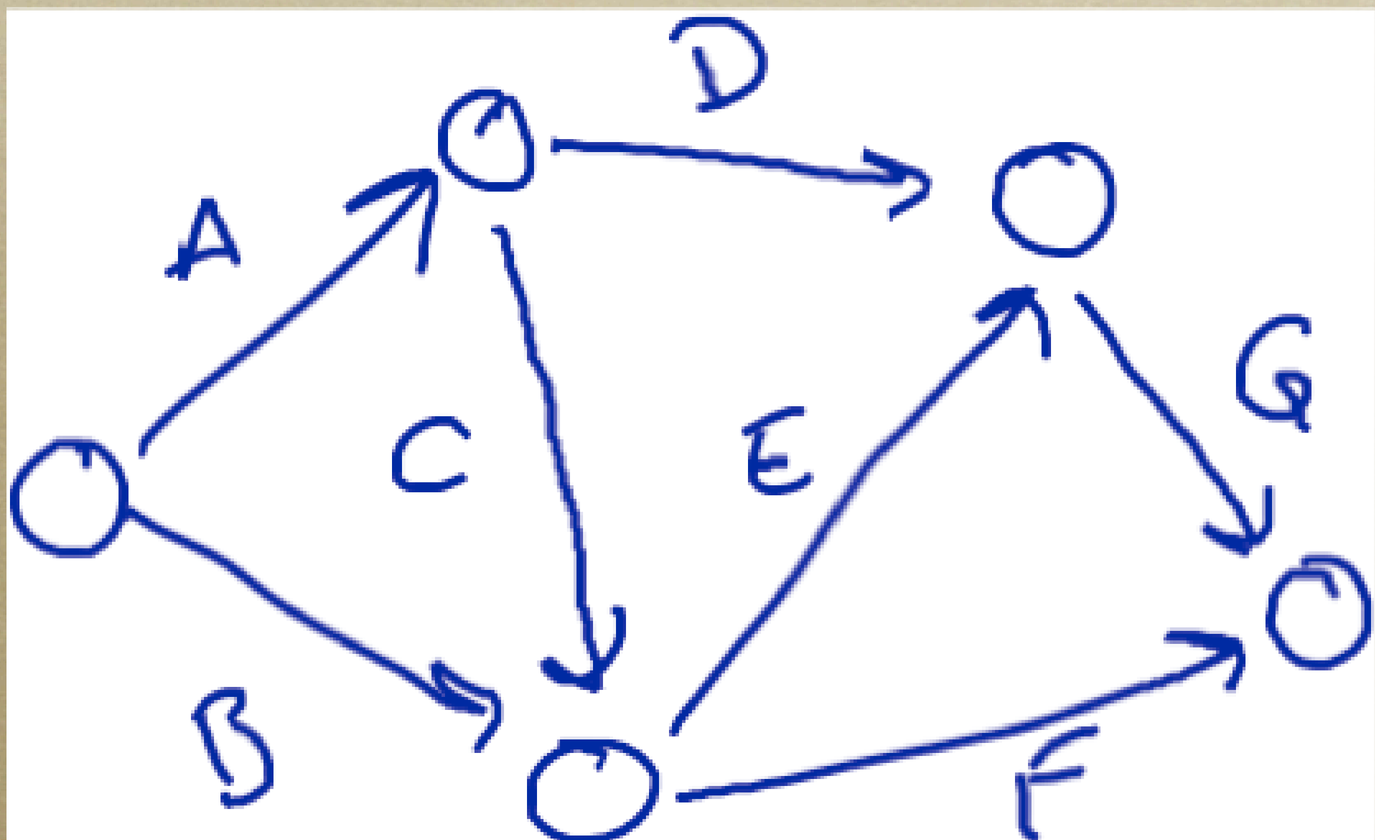
- *Guarantee Pareto in play against a wide variety of learners, but still guarantee safety against arbitrary learner and best-response against fixed learner*
- *Do so in arbitrary games (not just repeated matrix games)*

Structured games

A structured game



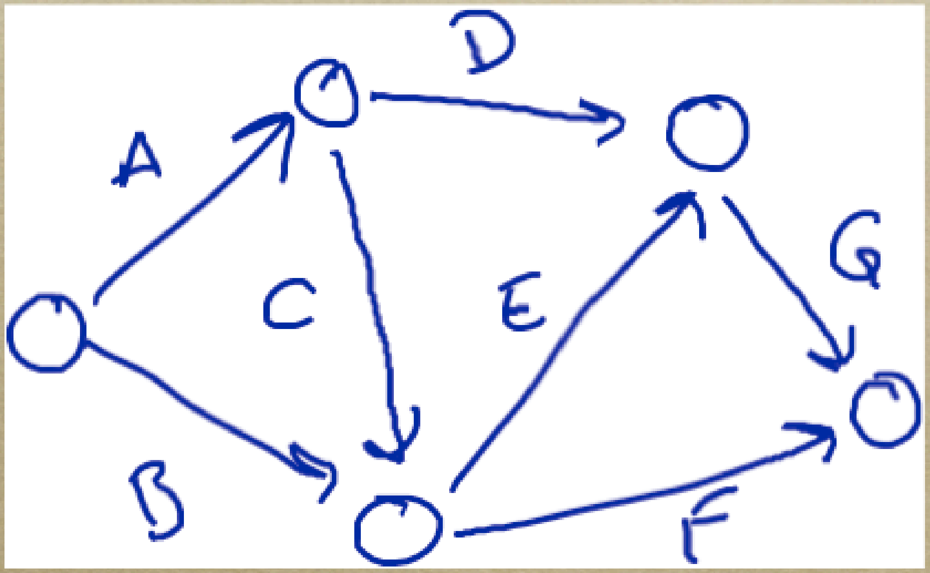
Abstract view



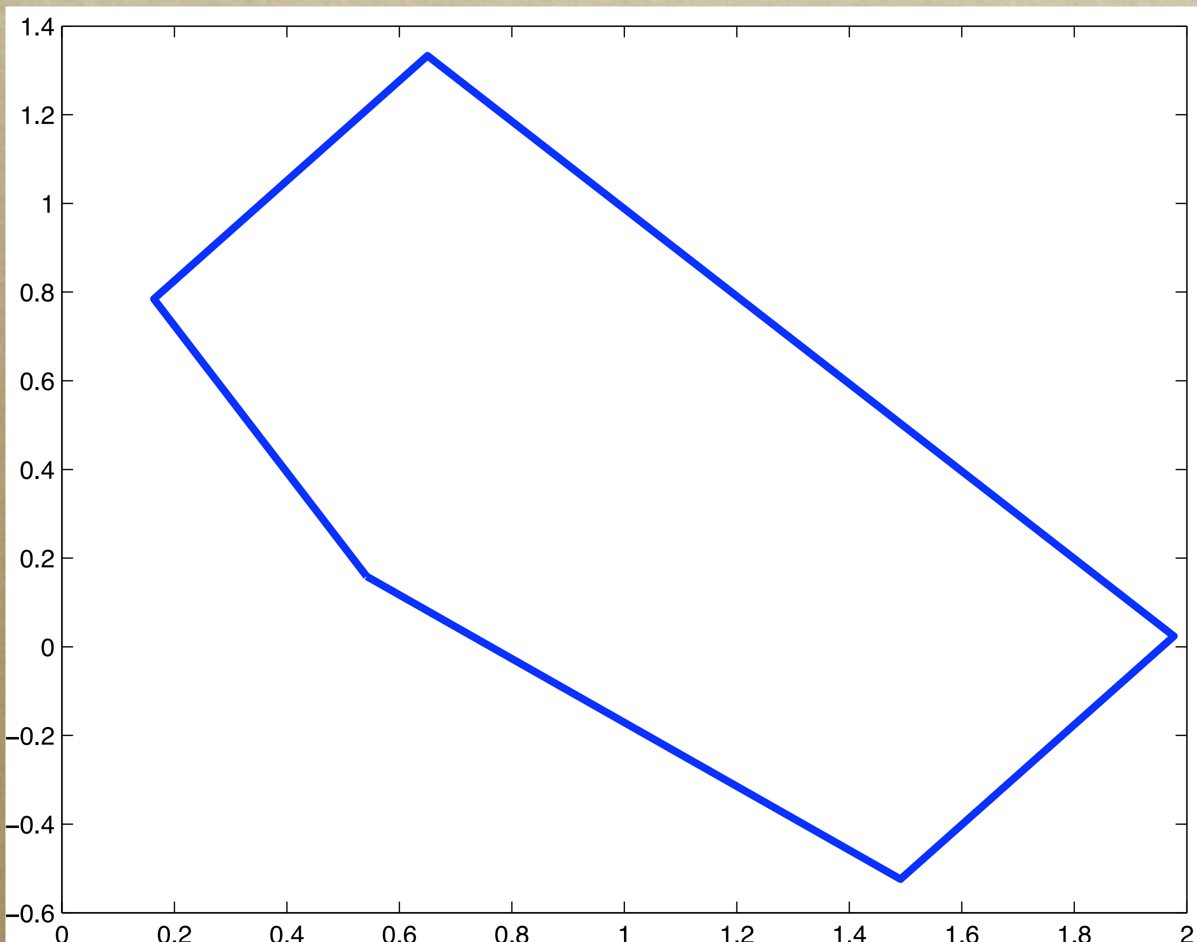
Feasible plays are convex, payoffs linear

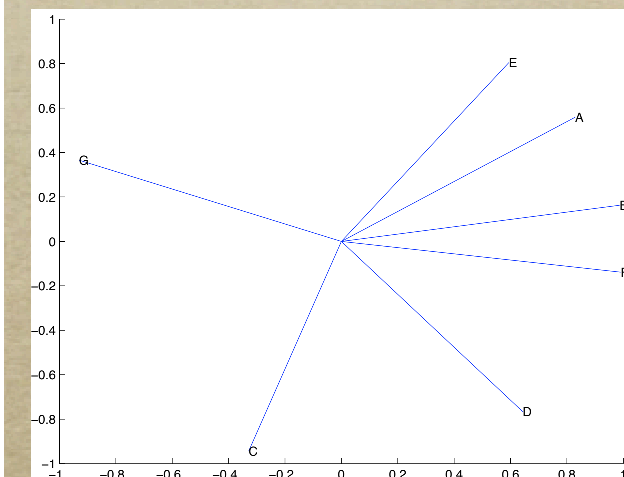
$$payoff = c_A A + c_B B + c_C C + c_D D + c_E E + c_F F + c_G G$$

- $A+B = 1$
- $C+D = A$
- $E+F = B+C$
- $G = D+E$
- $F+G = 1$
- $A,B,C,D,E,F,G \geq 0$



Feasible region





coordinate axes

OCP

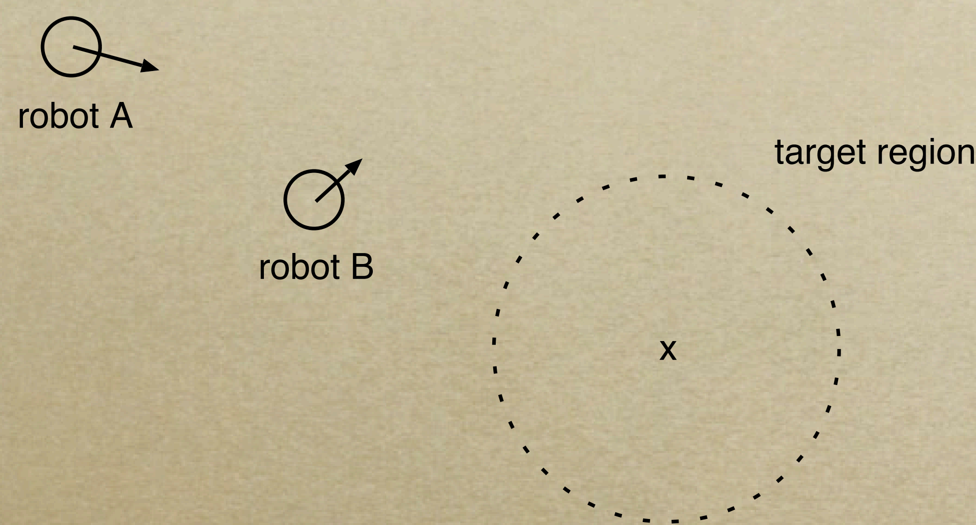
- *Learning problem is an online convex program: known convex feasible region, unknown but linear costs depend on actions of other players*
- *Lots of other games can be represented this way too*
  - *e.g., extensive-form games like poker*
- *Matrix game version exponentially bigger*

Scaling up

Playing realistic games

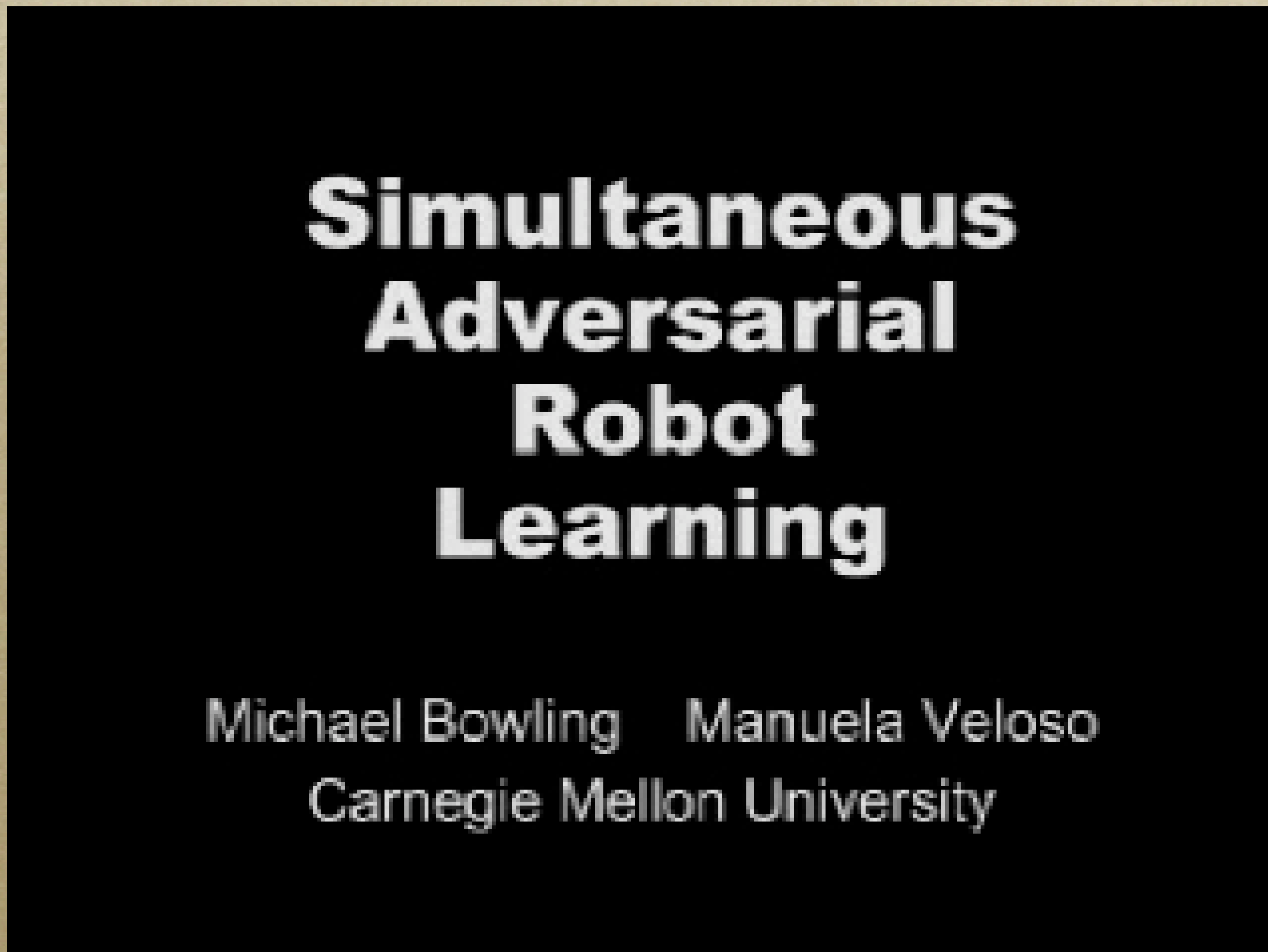
- *Main approaches*
  - *Non-learning*
  - *Opponent modeling*
    - *as noted above, guarantees are slim*
  - *Policy gradient*
    - *usually not a version with no regret*
  - *Growing interest in no-regret algorithms, but fewer results so far*

Policy gradient example



- *Keep-out game: A tries to get to target region, B tries to interpose*
- *Subproblem of RoboCup*

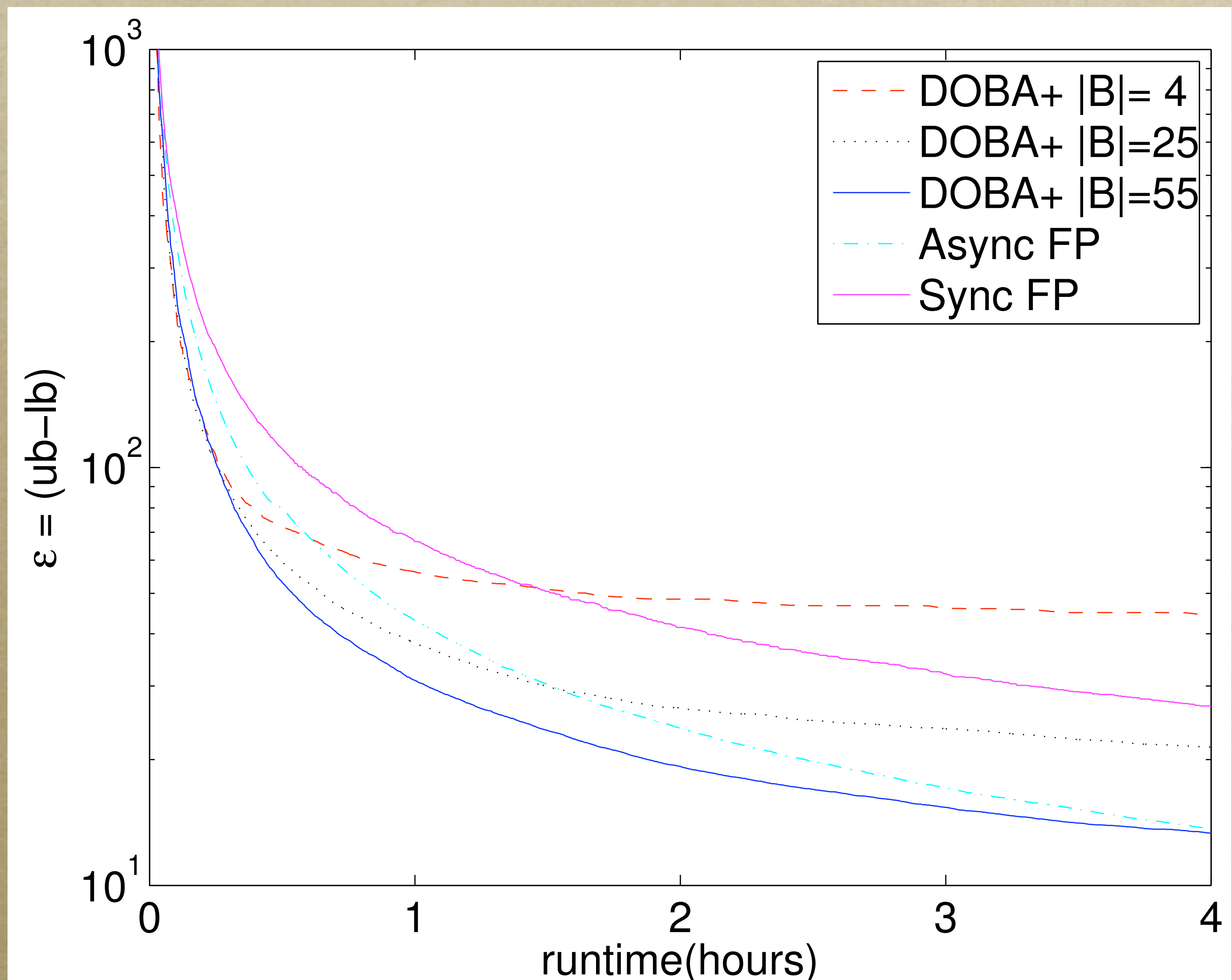
Policy gradient example



No-regret example

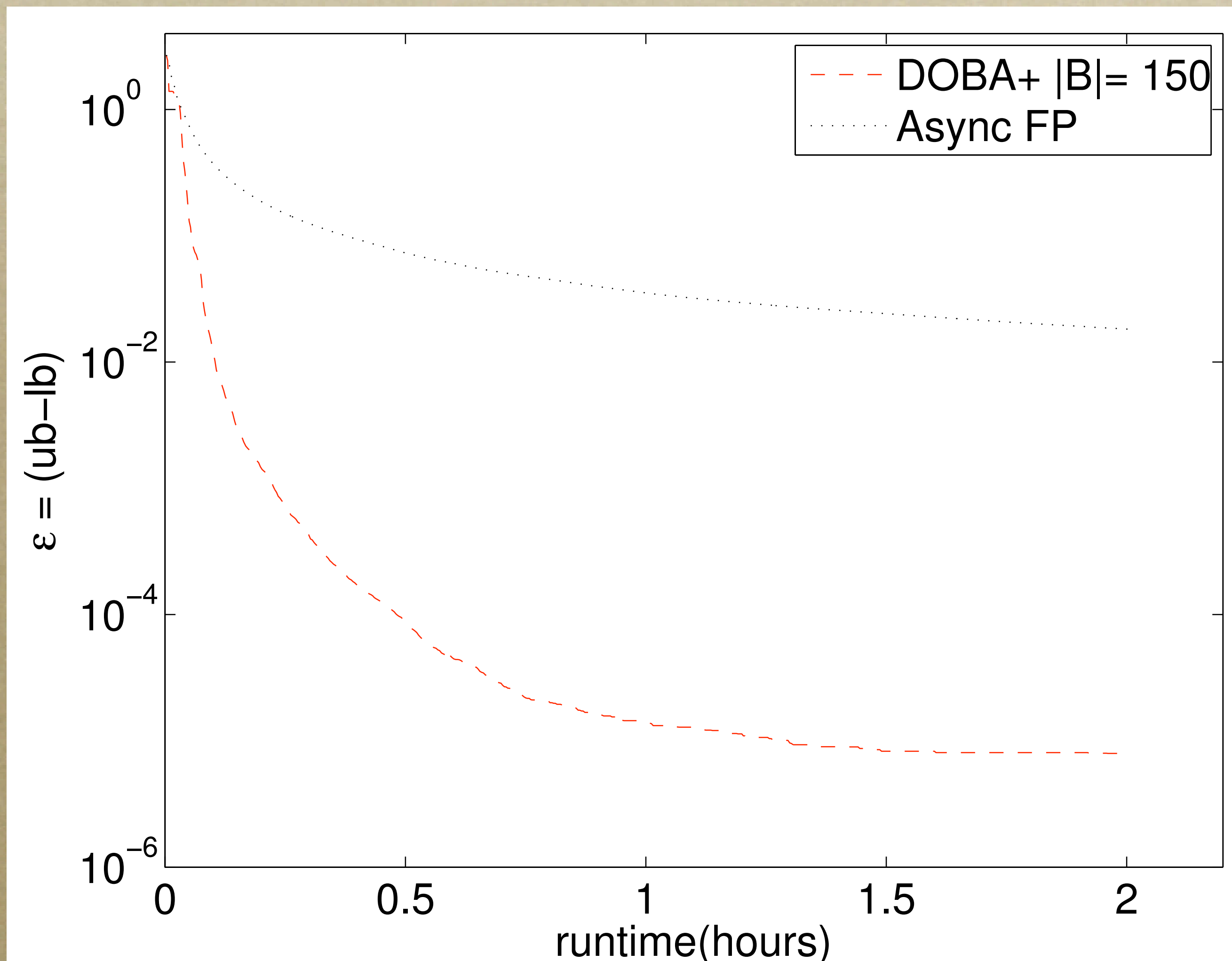
- *Learning to play poker from experience*
- *Play two learners head-to-head*
- *Results you're about to see are from Brendan McMahan's thesis*
- *They are from an algorithm that is some-regret; no-regret experiments in progress*

RI Hold'Em



DOBA+ |B|= 4
DOBA+ |B|=25
DOBA+ |B|=55
Async FP
Sync FP
ε = (ub−lb)
runtime(hours)
10^3
10^2
10^1
0
1
2
3
4

Texas Hold'Em



No-regret algorithm; RI Hold'em