15-780: Graduate AI
Computational Game Theory

Geoff Gordon (this lecture)
Ziv Bar-Joseph
TAs Geoff Hollinger, Henry Lin

Admin

- *HW5 out today (due 12/6—last class)*
- *Project progress reports due 12/4*
  - *One page: accomplishments so far, plans, problems, preliminary figures, …*
- *Final poster session: Thursday, 12/13, 5:30–8:30PM, NSH Atrium*
  - *Final reports due at poster session*

Games and AI

Why games?

- *Economics*
- *Organizations*
- *Warfare*
- *Recreation*

Why games?

- *Economics*
  - *FCC spectrum auctions, Google/Yahoo ad placement, supply chains, stock market, …*
- *Organizations*
- *Warfare*
- *Recreation*

Why games?

- *Economics*
- *Organizations*
  - *formation of official / actual chains of command in businesses, governments, armies, …*
- *Warfare*
- *Recreation*

Why games?

- *Economics*
- *Organizations*
- *Warfare*
    - *dogfights, sensor tasking, troop deployment, logistics, settlement negotiations …*
- *Recreation*

Why games?

- *Economics*
- *Organizations*
- *Warfare*
- *Recreation*
  - *chess, go, poker, football, …*

Problems to solve

- *Help agents choose good strategies*
  - *play poker well, find the hidden tanks*
- *Design games w/ desired properties*
  - *e.g., an auction that maximizes revenue*
- *Predict what humans will do*
  - *esp. as part of a complex system*

Recap

	A	*U*
A	*3, 4*	*0, 0*
U	*0, 0*	*4, 3*

- *Matrix games*
  - *2 or more players choose action simultaneously*
  - *Each from discrete set of choices*
  - *Payoff to each is function of all agents' choices*

Recap

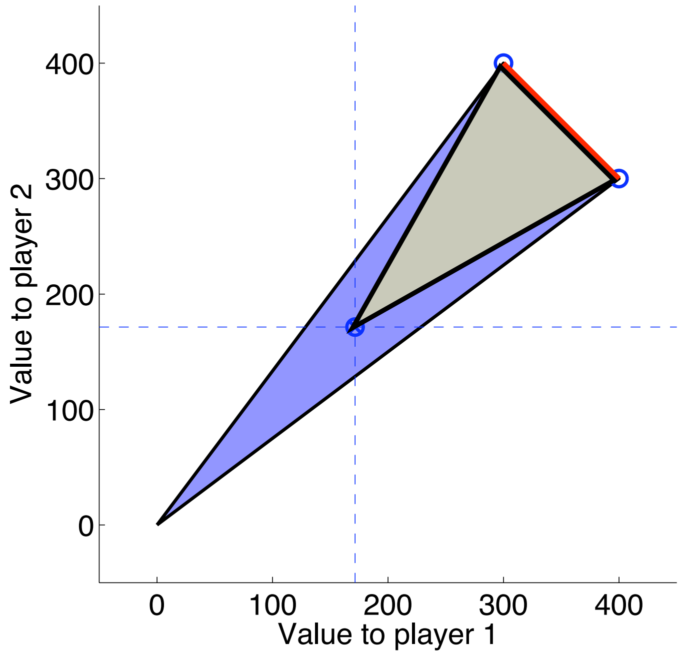
- ***Safety value*** *is best I can guarantee myself with worst-case opponent*
- *All we need to know if zero-sum or paranoid*
- *If we assume more about opponent (e.g., rationality) we might be able to get more reward*

Recap

- *Equilibrium = distribution over joint strategies so that no one agent wants to deviate unilaterally*
    - *Minimax: only makes sense in zero-sum two-player games, easy to compute*
    - *Nash: independent choices, the equilibrium everyone talks about*
    - *Correlated: uses moderator*

Recap

- *Pareto dominance: not all equilibria are created equal*
- *For any in brown triangle, there is one on red line that's at least as good for **both** players*
- *Red line = **Pareto dominant***



Choosing strategies

Choosing good strategies

- *Three fundamentally different cases:*
  - *one-shot*
  - *one-shot w/ communication*
  - *repeated*

One-shot game

- *One-shot = play game once, never see other players before or after*
- *What is a good strategy to pick in a one-shot game?*
    - *e.g., Lunch*

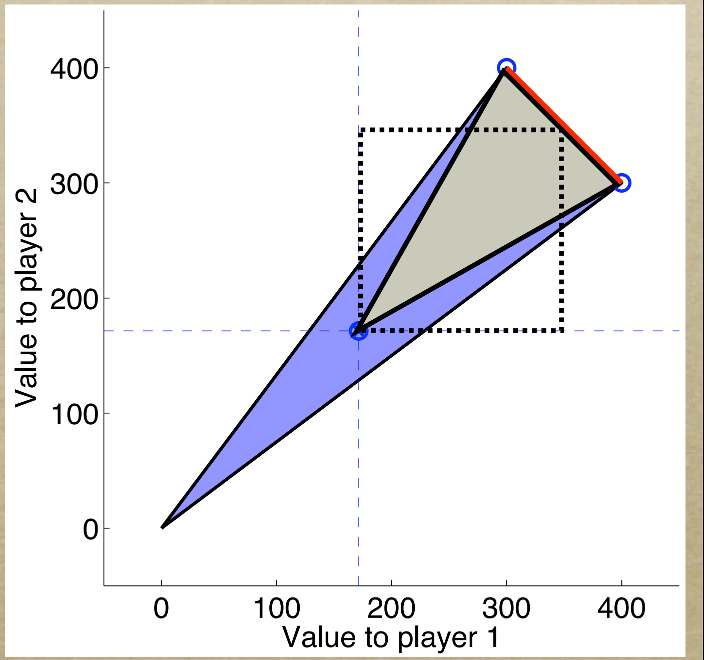
	A	*U*
A	*3, 4*	*0, 0*
U	*0, 0*	*4, 3*

One-shot game

- *Answer: it was a trick question*
- *No matter what we play, there's no reason to believe other player will play same*
- *Called the* ***coordination problem***

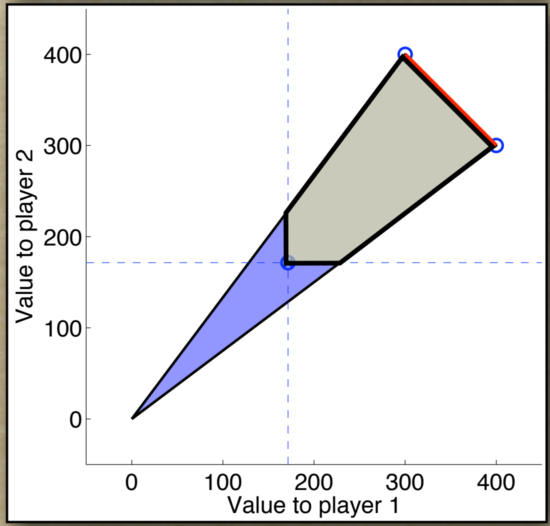
One-shot + communication

- *One-shot w/o comm is boring*
- *If comm allowed, **designer** could tell all players an equilibrium, and **moderator** could implement it*
- *E.g., "flip a coin" CE*
- *Can simulate moderator; what about designer?*



One-shot + communication

- *To replace designer, players could **bargain***
- *Problems:*
  - *predict what will happen in case of disagreement*
  - *incomplete information*
  - *world state*



Repeated games

- *One-shot w/ comm motivates need to compute equilibria—will discuss next*
- *Repeated case will motivate learning—more later*

Computing equilibria

Computing equilibria

- *A central problem of complexity theory*
- *Different answers depending on type of equilibrium desired*

How hard is it to find Nash?

- *At border of poly-time computability*
- *No poly-time algorithms known*
  - *even for 2-player games w/ 0/1 payoffs*
  - *results (since 2004) of Papadimitriou, Chen & Deng, Abbott et al*
- *Easy to find in nondeterministic poly-time*

How hard is it to find Nash?

- *Interestingly, adding almost any interesting restriction makes the problem NP-complete*
- *E.g., existence of Nash w/ total payoff $\geq k$ is NP-complete*

How hard is it to find CE?

- *Finding CE = solving LP*
- *Size = O(size(payoff matrices) actions^2)*
- *So, finding CE is poly-time*
  - *as is optimizing sum of payoffs*
- *E.g., 3-player, 10-action game: 271 constraints, 10^3 variables, sparsity ~10%*

But…

- *But, size of payoff matrices exponential in number of players*
- *So, not practical to write down a matrix game with millions of players, much less find CE*
- *Seems unsatisfying…*

Succinct games

- *In a* ***succinct game****, payoff matrices are written compactly*
- *E.g., a million people sit in a big line*
- *Each chooses +1 or –1*
- *If I choose X, left neighbor chooses L, and right neighbor chooses R, my payoff is*
  - $XL - XR$

CE in succinct games

- *Finding equilibria is harder in succinct games: can't afford to write out payoff matrices or LP*
- *But, can find CE in poly time in large class of succinct games: clever algorithm due to*

  Christos H. Papadimitriou. Computing Correlated Equilibria in Multi-Player Games. STOC 37, 2005.
- *Interestingly, highest-total-payoff CE is NP-hard*

Summary of complexity

- *Nash: border of poly-time*
  - *even in 2-player 0/1 case*
- *CE: poly-time*
  - *highest-payoff CE: poly-time*
- *Succinct CE: poly-time*
  - *highest-payoff sCE: NP-hard*

Finding CE

Recap: finding CE

	A	*U*
A	*a*	*b*
U	*c*	*d*

	A	*U*
A	*4,3*	*0*
U	*0*	*3,4*

Row recommendation A $4a + 0b \geq 0a + 3b$

Row recommendation U $0c + 3d \geq 4c + 0d$

Col recommendation A $3a + 0c \geq 0a + 4c$

Col recommendation U $0b + 4d \geq 3b + 0d$

$a, b, c, d \geq 0 \qquad a + b + c + d = 1$

Interpretation

	A	U
A	a	b
U	c	d

	A	U
A	4,3	0
U	0	3,4

- *Row reward is 4a + 0b + 0c + 3d*
- *What if, whenever moderator tells us A, we play U instead?*

Interpretation

	A	U
A	a	b
U	c	d

	A	U
A	4,3	0
U	0	3,4

- *Row reward is 4a + 0b + 0c + 3d*
- *… becomes 0a + 3b + 0c + 3d*
- *Difference 4a – 3b is* ***regret*** *for switch*
  - *+ve bad, –ve good*

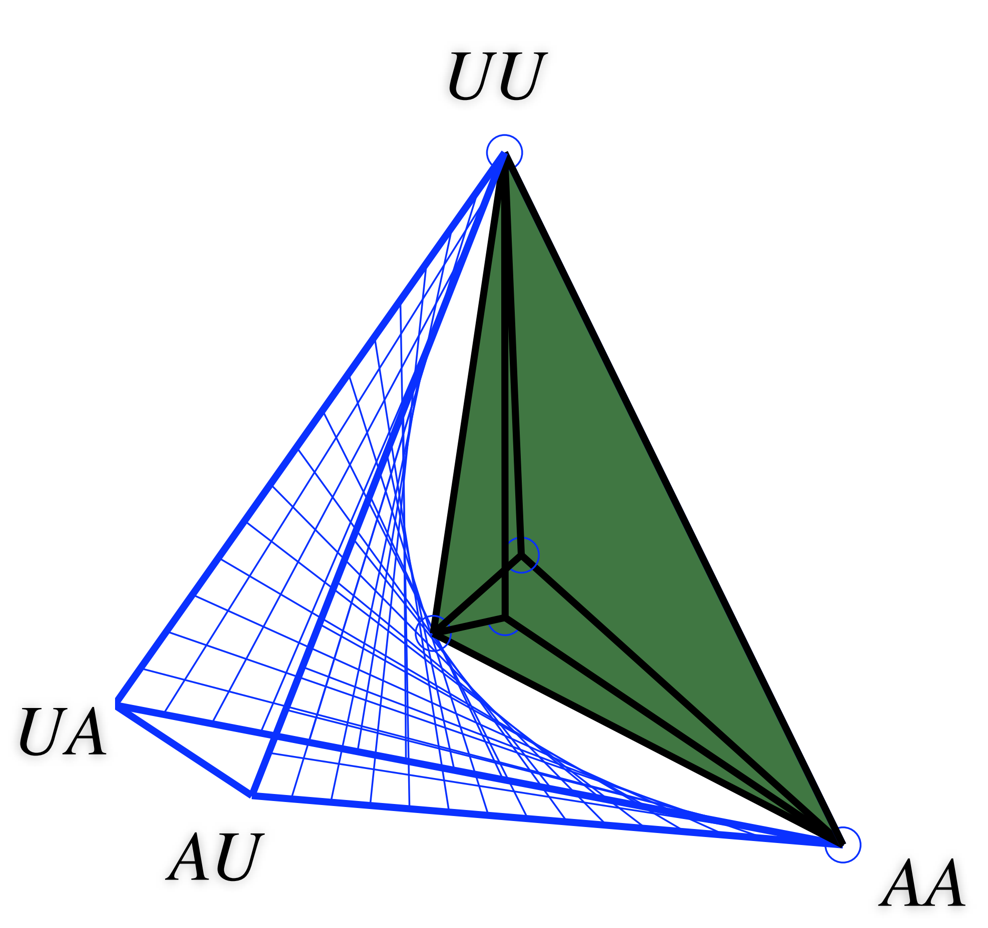
Interpretation

	A	U
A	a	b
U	c	d

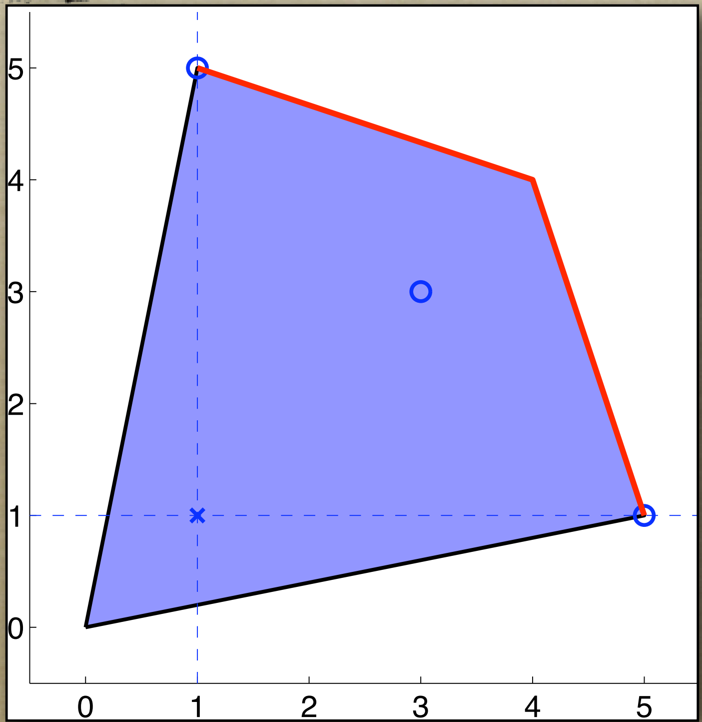
	A	U
A	$4,3$	0
U	0	$3,4$

- *Difference $4a - 3b$ is **regret** for $A \to U$*
- *Constraint $4a - 3b \geq 0$ means we don't want to switch $A \to U$*
- *Other constraints: we don't want $U \to A$, Col doesn't want $A \to U$ or $U \to A$*

CE: the picture



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)*

Finding Nash

Shapley's game

	A	*B*	*C*
1	*0,0*	*1,0*	*0,1*
2	*0,1*	*0,0*	*1,0*
3	*1,0*	*0,1*	*0,0*

Support enumeration algorithm

	A	B	C
1	0,0	1,0	0,1
2	0,1	0,0	1,0
3	1,0	0,1	0,0

- *Enumerate all support sets for each player*
- *Row: 1, 2, 3, 12, 13, 23, 123*
- *Col: A, B, C, AB, AC, BC, ABC*
- *$7 \times 7 = 49$ possibilities*

Support enumeration

- *For each pair of supports, solve an LP*
- *Vars are P(action) for each action in support (one set for each player), and also expected value to each player*
- *Constraints:*
  - *All actions in support have value v*
  - *All not in support have value ≤ v*
  - *Probabilities in support ≥ 0, sum to 1*

Support enumeration

	A	B	C
1	0,0	1,0	0,1
2	0,1	0,0	1,0
3	1,0	0,1	0,0

- *Checking singleton supports is easy: sum-to-1 constraint means p=1 for action in support*
- *So just check whether actions out of support are worse*

Try 2-strategy supports: 12, AB

	A	*B*	*C*
1	*0,0*	*1,0*	*0,1*
2	*0,1*	*0,0*	*1,0*
3	*1,0*	*0,1*	*0,0*

- *Payoff of Row 1: 0 p(A) + 1 p(B) = v*
- *Payoff of Row 2: 0 p(A) + 0 p(B) = v*
- *Payoff of Col A: 0 p(1) + 1 p(2) = w*
- *Payoff of Col B: 0 p(1) + 0 p(2) = w*

Try 2-strategy supports: 12, AB

	A	*B*	*C*
1	*0,0*	*1,0*	*0,1*
2	*0,1*	*0,0*	*1,0*
3	*1,0*	*0,1*	*0,0*

- $0\,p(A) + 1\,p(B) = v = 0\,p(A) + 0\,p(B)$
- $0\,p(1) + 1\,p(2) = w = 0\,p(1) + 0\,p(2)$
- *Row payoff* $\geq$ *row 3:* $v \geq 1\,p(A) + 0\,p(B)$
- *Col payoff* $\geq$ *col C:* $w \geq 1\,p(1) + 0\,p(2)$

More supports

- *Other 2-vs-2 are similar*
- *We also need to try 1-vs-2, 1-vs-3, and 2-vs-3, but in interest of brevity: they don't work either*
- *So, on the 49th iteration, we reach 123 vs ABC…*

123 vs ABC

	A	*B*	*C*
1	*0,0*	*1,0*	*0,1*
2	*0,1*	*0,0*	*1,0*
3	*1,0*	*0,1*	*0,0*

- Row 1: $0\, p(A) + 1\, p(B) + 0\, p(C) = v$
- Row 2: $0\, p(A) + 0\, p(B) + 1\, p(C) = v$
- Row 3: $1\, p(A) + 0\, p(B) + 0\, p(C) = v$
- So, $p(A) = p(B) = p(C) = v = 1/3$

123 vs ABC

	A	B	C
1	0,0	1,0	0,1
2	0,1	0,0	1,0
3	1,0	0,1	0,0

- Col A: $0\,p(1) + 0\,p(2) + 1\,p(3) = w$
- Col B: $1\,p(1) + 0\,p(2) + 0\,p(3) = w$
- Col C: $0\,p(1) + 1\,p(2) + 0\,p(3) = w$
- So, $p(1) = p(2) = p(3) = w = 1/3$

Nash of Shapley

- *There are nonnegative probs p(1), p(2), & p(3) for Row that equalize Col's payoffs for ABC*
- *There are nonnegative probs p(A), p(B), & p(C) for Col that equalize Row's payoffs for 123*
- *No strategies outside of supports to check*
- *So, we've found the (unique) NE*

Learning in Games

Repeated games

- *One-shot games: important questions were equilibrium computation, coordination*
- *If we get to play many times, **learning** about other players becomes much more important than static equilibrium-finding*
- *Equilibrium computation, coordination can be achieved as a by-product of learning*

Learning

- *Start with beliefs / inductive bias about other players*
- *During repeated plays of a game*
  - *or during one long play of a game where we can revisit the same or similar states*
- *Adjust our own strategy to improve payoff*

Rules of game

- *In addition to learning about other players, can learn about rules of game*
- *Important in practice, but won't talk about it here*
- *Many of the algorithms we'll discuss generalize straightforwardly*

Learning and equilibrium

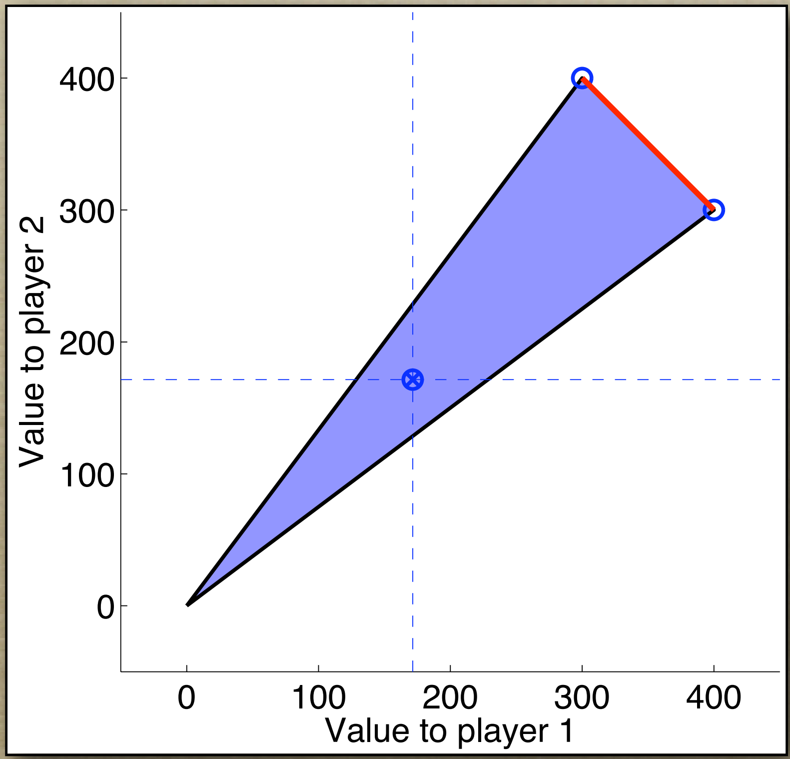
- *Equilibrium considerations place constraints on learning algorithms*
- *At the least, if all players "rational," would hope outcome of learning is near an equilibrium in the limit*
    - *Else some player would want to use a different learning algorithm*
- *E.g., wouldn't expect consistent excess of R*

Equilibria in repeated games

- *Possible confusion: equilibria in repeated games can be much more complicated than in stage game*
- *Complicated equilibria are (unfortunately) the relevant ones*

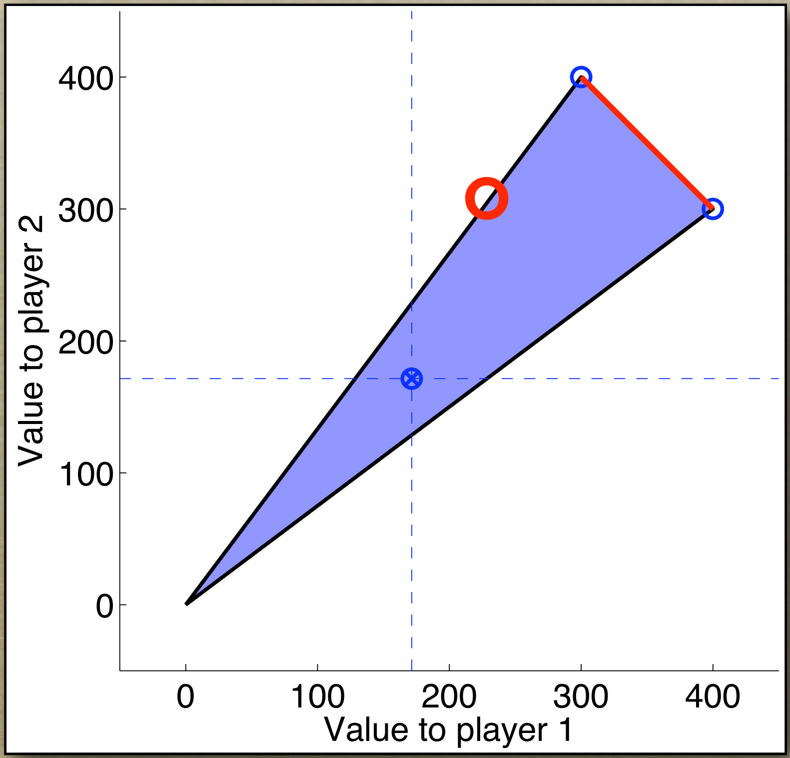
E.g., Lunch

- *In one-shot Lunch game, 3 NE*



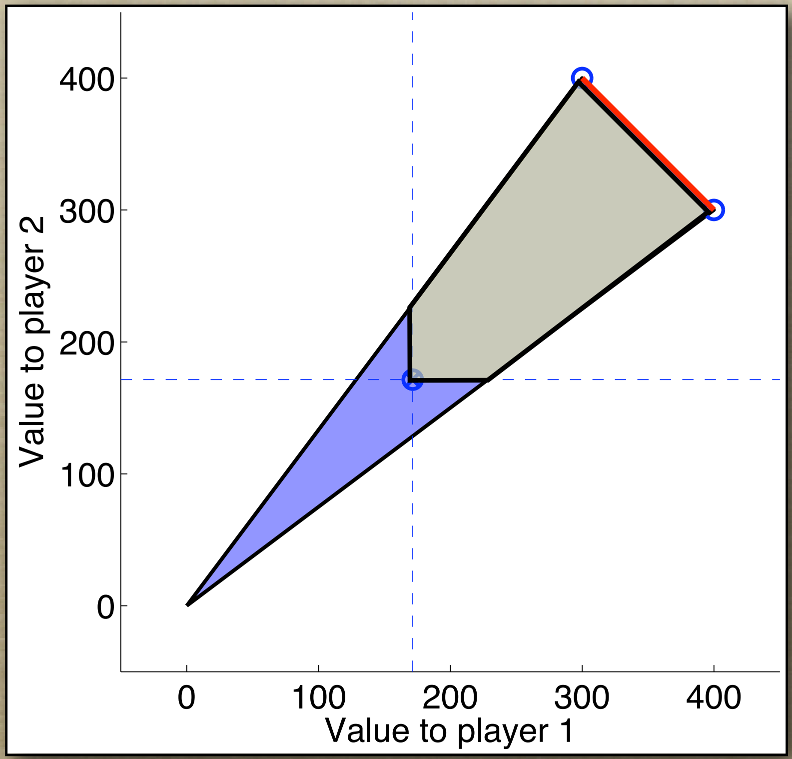
E.g., Lunch

- *In repeated game, for example:*
- *We'll both go to Ali Baba 6 times, then to different places 2 times, then repeat. You'd better do what I say, or else I'll make sure you get the least possible payoff.*



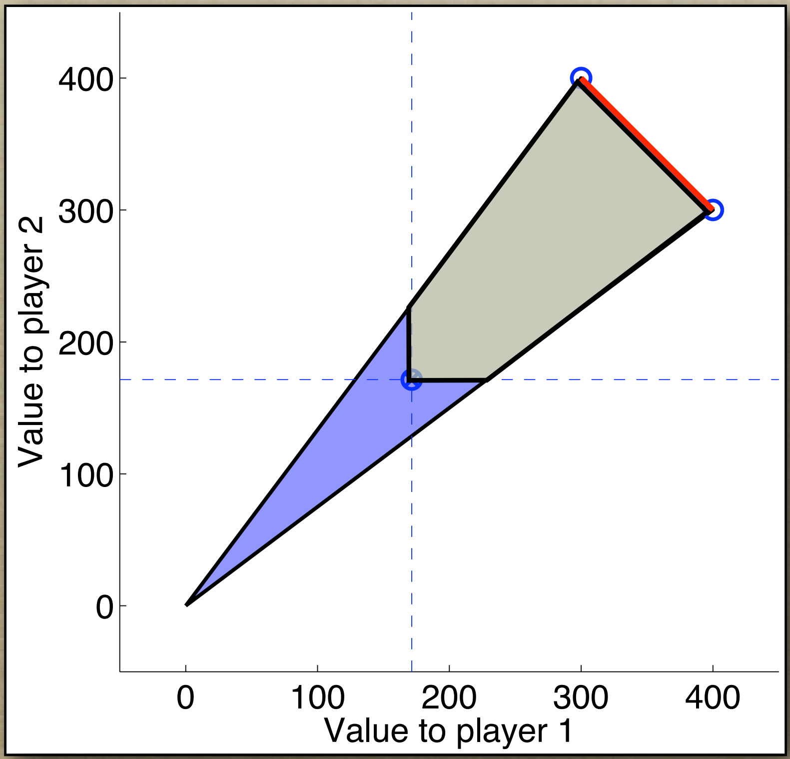
Folk Theorem

- *In fact,* ***any*** *feasible payoff that is above safety values corresponds to some Nash equilibrium*
- *Makes designing and analyzing learning algorithms difficult…*



Bargaining

- *I'd like AA best*
- *And nobody wants to converge to interior of pentagon*
- *"Steering" outcome of learning is an important open question*



Opponent modeling

First try

- *Run any standard supervised learning algorithm to predict*
  - *payoff of each of my actions, or*
  - *play of all other players*
- *Now act to maximize my predicted utility on next turn*

Fictitious play

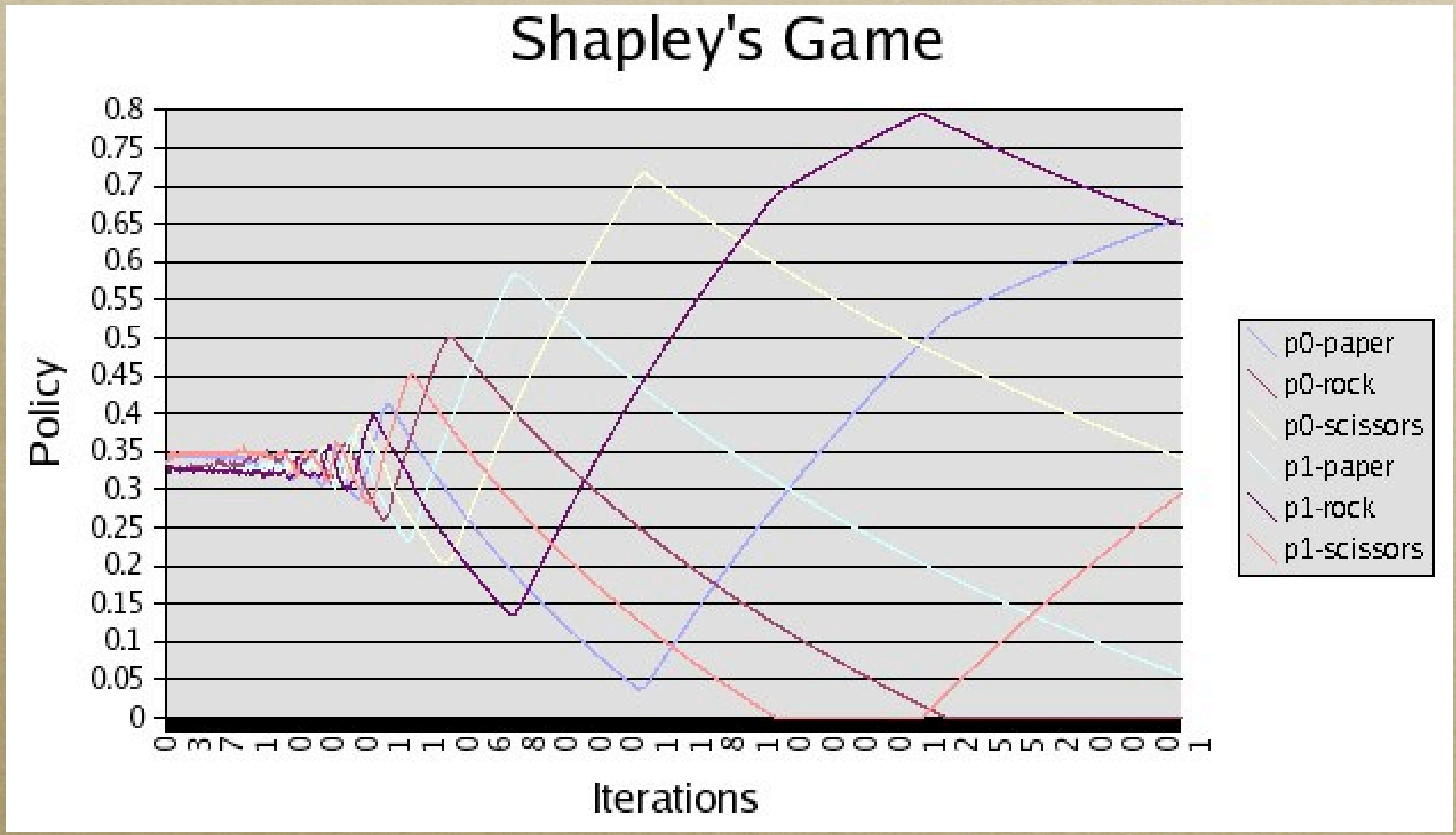
- *For example, count up number of times opponent played Rock, Paper, or Scissors*
- *If Rock is highest, play Paper, etc.*
- *This algorithm is called* ***fictitious play***

Shapley's game

	R	*P*	*S*
R	*0,0*	*1,0*	*0,1*
P	*0,1*	*0,0*	*1,0*
S	*1,0*	*0,1*	*0,0*

non-zero-sum version of rock, paper, scissors

Fictitious play



- *Even in self-play, FP can do badly*

Fictitious play

- *Worse yet, what if opponent knows we're using FP?*
  - *We will lose every time*

Second try

- *We were kind of short-sighted when we chose to optimize our immediate utility*
- *What if we formulate a prior, not over single plays, but over (infinite) sequences of play (conditioned on our own strategy)?*
- *E.g., P(7th opp play is R, 12th is S | my first 11 plays are RRRPRPRSSSR) = 0.013*

Rational learner

- *Now we can look ahead: find best play considering all future effects*
- *R might garner more predicted reward now, but perhaps S will confuse opponent and let me get more reward later…*
- *This is called* ***rational learning***
- *A complete rational learner must also specify tie-break rule*

Rational learner: discussion

- *First problem: maximization over an uncountable set of strategies*
- *Second problem: our play is still deterministic, so if opponent gets a copy of our code we're still sunk*
- *What if we have a really big computer and can hide our prior?*

Theorem

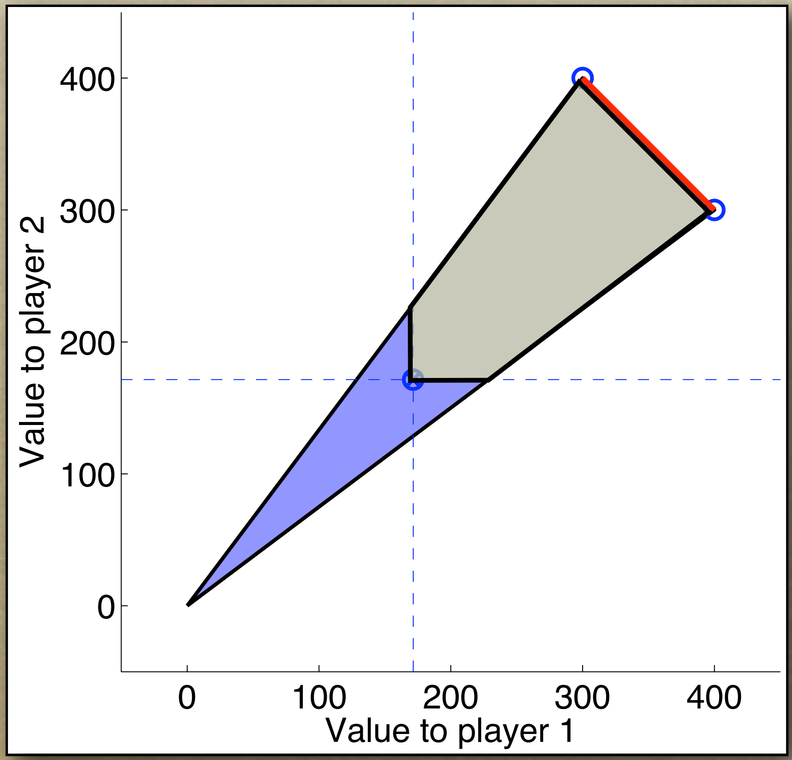
- *Any vector of rational learners which (mumble mumble) will, when playing each other in a repeated game, approach the play frequencies and payoffs of some Nash equilibrium arbitrarily closely in the limit*

Ehud Kalai and Ehud Lehrer. Rational Learning Leads to Nash Equilibrium. Econometrica, Vol. 61, No. 5, 1993.

What does this theorem tell us?

- *Problem: “mumble mumble” actually conceals a condition that’s difficult to satisfy in practice*
  - *for example, it was violated when we peeked at prior and optimized response*
  - *nobody knows whether there’s a weaker condition that guarantees anything nice*

What does this theorem tell us?



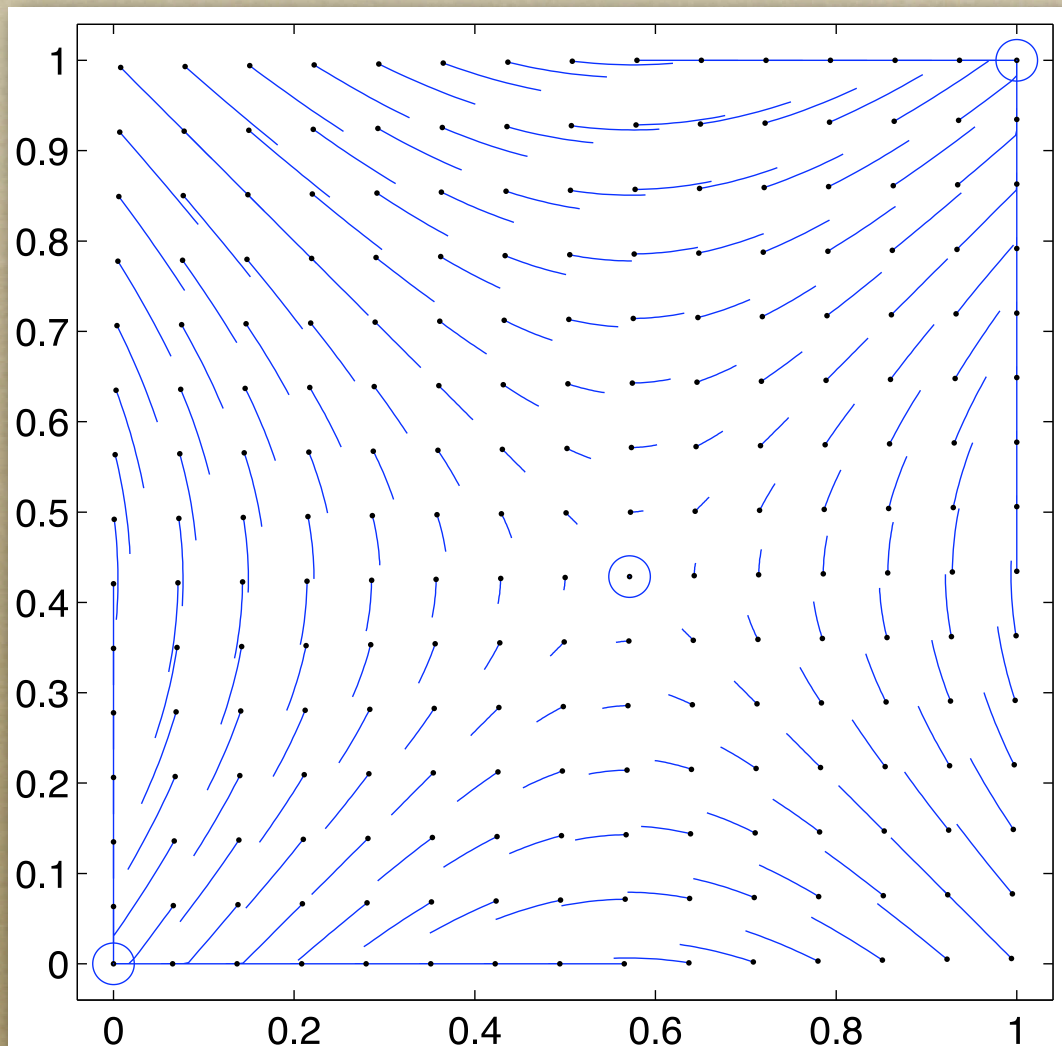
- *And, as mentioned above, there are often a lot of Nash equilibria*

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

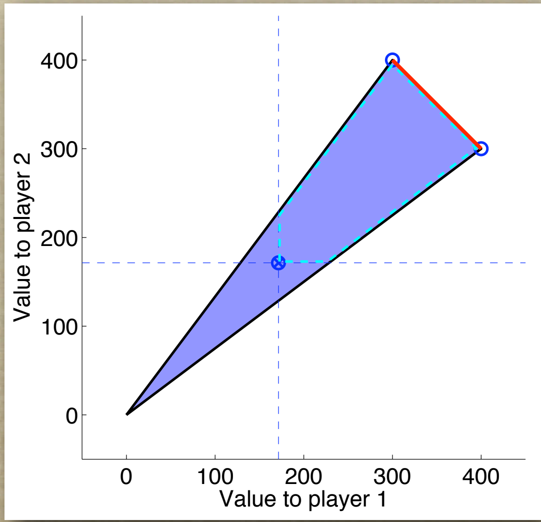
Theorem

- *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 (its part 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

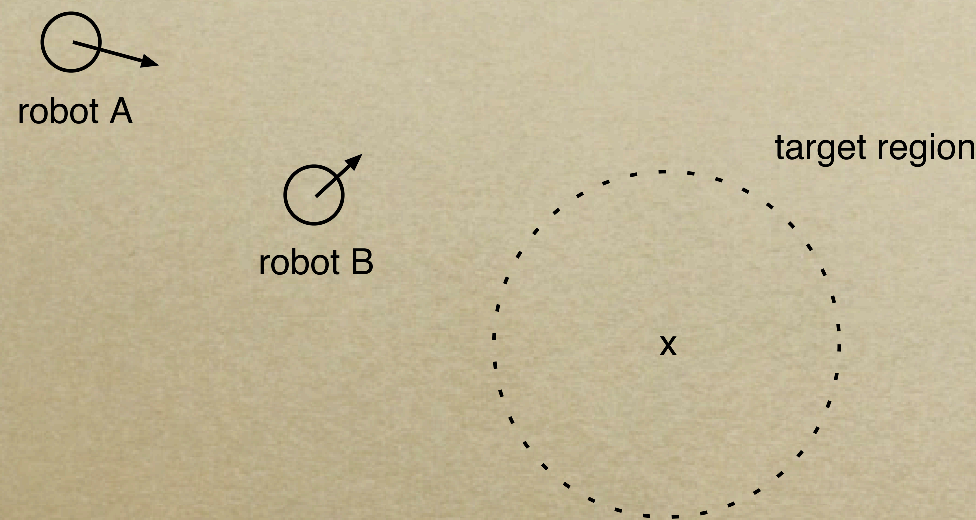
- *First learning algorithm we've discussed that guarantees Pareto in self-play*
- *Only a few algorithms with this property so far, all since about 2003 (Brafman & Tennenholtz, Powers & Shoham, Gordon & Murray)*
- *Can't really claim it's "bargaining"— would like to be able to guarantee something about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

Scaling up

Playing realistic games

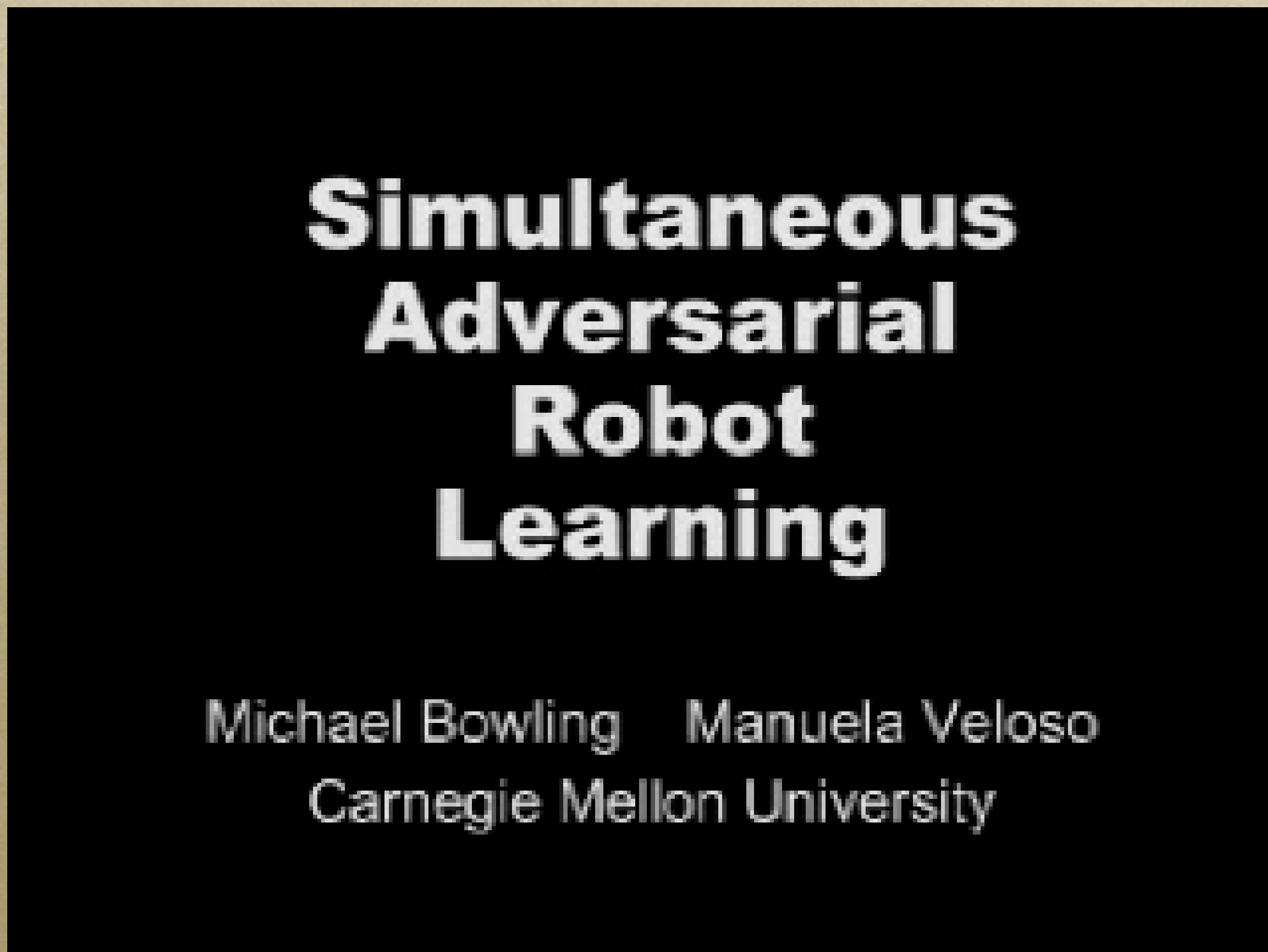
- *Main approaches*
  - *Non-learning*
  - *Opponent modeling*
    - *as noted above, guarantees are slim*
  - *Policy gradient*
    - *usually not a version with no regret*
  - *Growing interest in no-regret algorithms, but fewer results so far*

Policy gradient example



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

Policy gradient example



Mechanism design

Note: we didn't get to the remaining slides in class

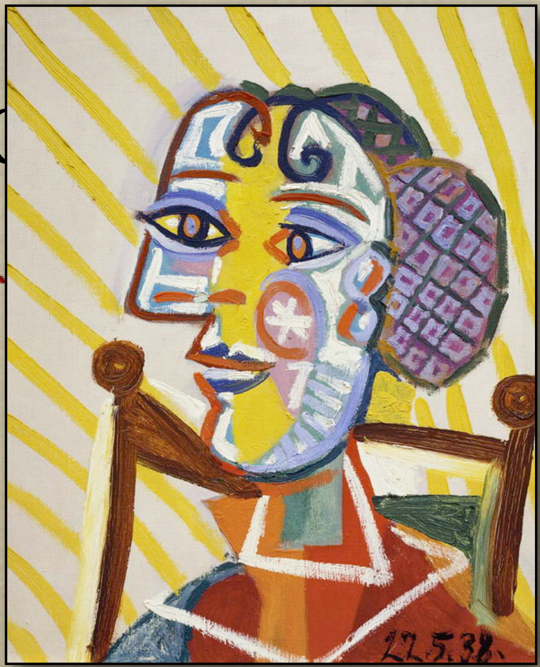
Mechanism design

- *Recall: want to design a game that has desired properties*
- *E.g., want equilibrium to have highest possible total payoff for players, or want game designer to profit as much as possible*

Social choice

- *Group of players must jointly select an outcome x*
- *Player i has payoff $R_i(x, w_i)$*
  - *w_i is a random signal, known only to player i, called **type***
- *If we knew all the w_i values, could choose*
  - $x = \arg\max_x \boxed{\sum_i R_i(x, w_i)}$ ← social welfare
- *But players aren't motivated to reveal w_i*

Example: allocatio



- *Choose which player gets a valuable, indivisible item*
- *Each player has private value w_i*
- *Social welfare maximized by giving item to player with highest valuation*
- *So, everyone wants to say "it's worth \$100M to me!!!"*

Example: auction

- *For allocation problem, can fix overbidding problem by requiring players to pay according to their bids*
- *E.g., highest bidder gets item, pays bid price ("first price auction")*

Mechanism

- *This is a simple example of a **mechanism**: a game which determines social choice x as well as payments to/from players*
- *Actions = bids*
- *Strategy = (type $\mapsto$ bid)*

Problem

- *First-price auction mechanism has a problem*
- *Players will lie and say item is worth less than they think it is*
- *Might cause suboptimal allocation (but only if players don't know correct distribution over others' valuations)*

Is there a general solution?

- *Want:*
  - *mechanism implements socially optimal choice ("efficiency")*
  - *mechanism doesn't lose money ("budget balance")*

In general, no.

- *But we can do it for some social choice problems*
- *E.g.,* ***second-price*** *auction: highest bidder gets item, pays second-highest price*
- *Nobody wants to lie*
- *So, painting goes to player w/ high value*
- *And, mechanism always profits*

VCG

- *Second-price auction is example of Vickrey-Clarke-Groves mechanism*
- *Players tell mechanism their types (= valuations for all choices)*
- *Mechanism selects socially optimal outcome x^**
- *Payments determined according to VCG rule:*

VCG payment rule

- *Recall* x^* *= socially optimal outcome*
- *Define* x' *= outcome if player i absent*
- *Player i **receives** the sum of everyone else's reported valuations for* x^*
- *Player i **pays** the sum of everyone else's reported valuations for* x'

VCG rule

- *In allocation problem*
  - x^* *= item allocated to highest bidder*
  - *x' = item allocated to second bidder*
- *For winner:*
  - *sum of others' values in* $x^* = 0$
  - *sum of others' values in x' = 2nd bid*
- *For others: don't affect outcome, payment is 0*

More generally

- *Player i* ***receives*** *the sum of everyone else's reported valuations for x^**
- *Player i* ***pays*** *the sum of everyone else's reported valuations for x'*
- *… total payment for i is amount by which everyone else suffers due to i's presence — called* ***externality***