CISC453 Winter 2010

Making Complex Decisions Part B:
AIMA 3e Ch 17.5 - 17.7

Matthew Kelly
4mk44

Overview

- 17.5 - Game Theory
    - 17.5.1 - Single-move games
        - Two-finger Morra
        - Prisoner's Dillema
        - Domination, equilibrium
        - Maximin
    - 17.5.2 - Repeated games
        - Perpetual punishment
        - Tit-for-tat
    - 17.5.3 - Sequential games
        - Extensive form
- 17.6 - Mechanism Design
    - 17.6.1 - Auctions
    - 17.6.2 - Common goods

Game Theory

- Saw games in Ch 5.
  - Fully observable
  - Turn-taking
  - Minimax search
- Game theory
  - Partially observable
  - Multiple sources of partial observability
  - Perfect and imperfect information

Agent design

- One of the uses of game theory
- Use game theory to analyze and compute utility of possible decisions
    - Under the assumption that the other agents are also acting optimally...

- Example: two-finger Morra

Single-move games

- All players take an action simultaneously
- Defined by three components
  - Players
    - Two player and >2 players
    - Capitalized names like *O* , *Alice* , *John*
  - Actions
    - Lowercase names like *move* , *raise*
  - Payoff function
    - Gives utility to each player
    - For each set of actions
    - Two player games as matrix

Two-finger Morra - strategic/normal form

	O: one	O: two
E: one	E = +2 O = -2	E = -3 O = +3
E: two	E = -3 O = +3	E = +4 O = -4

More jargon...

- Strategy/policy
- Pure strategy
- Mixed strategy
  - For actions a , b and probability p
  - $[p : a ; (1 - p): b]$
    - i.e. [0.5:*one* ; 0.5:*two*]
- Strategy profile
- Outcome
- Solution as a "rational" strategy profile
  - Pure □**or** mixed

Prisoner's dilemma

- Two player, partially observable.
- Player can *testify* and serve 0 or 5 years
- Alternatively, *refuse* and serve 1 or 10 years

	Alice: testify	Alice: refuse
Bob: testify	A = -5 B = -5	A = -10 B = 0
Bob: refuse	A = 0 B = -10	A = -1 B = -1

Domination

- *testify* is a dominant strategy for the prisoner's dillema
- Strong domination
  - *s* strongly dominates *s′* if the outcome of *s* is better than the outcome of *s′* for □**all** other strategy profiles for others
- Weak domination
  - *s* weakly dominates *s′* if the outcome is better on at least one strategy profile and no worse on any other

Also need to define outcomes as,

- Pareto optimal
  - No other outcome that □**all** players would prefer
- Pareto dominated
  - By another outcome that □**all** players would prefer

Equilibrium

- We saw a dominant strategy equilibrium in the prisoner's dilemma

- In general, if:
  - No player can benefit by switching strategies
  - If every other player keeps the same strategy
- Then that strategy profile forms an equilibrium

Nash equilibrium

- Mathematician, Nobel prize winner in economics John Nash
- Proved every game has at least one equilibrium
  - i.e. a dominant strategy equilibrium
- In his honour, a general equilibrium is called a Nash equilibrium.

Back to the prisoners...

- What is the dilemma?
  - (testify, testify) is Pareto dominated by (refuse, refuse)

- Any way to reach (refuse, refuse)?
  - If we modify the game...
    - Change to repeated game
    - Add moral beliefs to change payoff function
    - Allow communication

More on dominant strategies

Consider the following game:

	Acme: bluray	Acme: dvd
Best: bluray	A = +9 B = +9	A = -4 B = -1
Best: dvd	A = -3 B = -1	A = +5 B = +5

Pure versus mixed strategy equilibrium

- Consider a pure strategy profile for two-finger Morra.
  - If the total fingers is even, O will want to switch
  - If it is odd, E will want to switch.

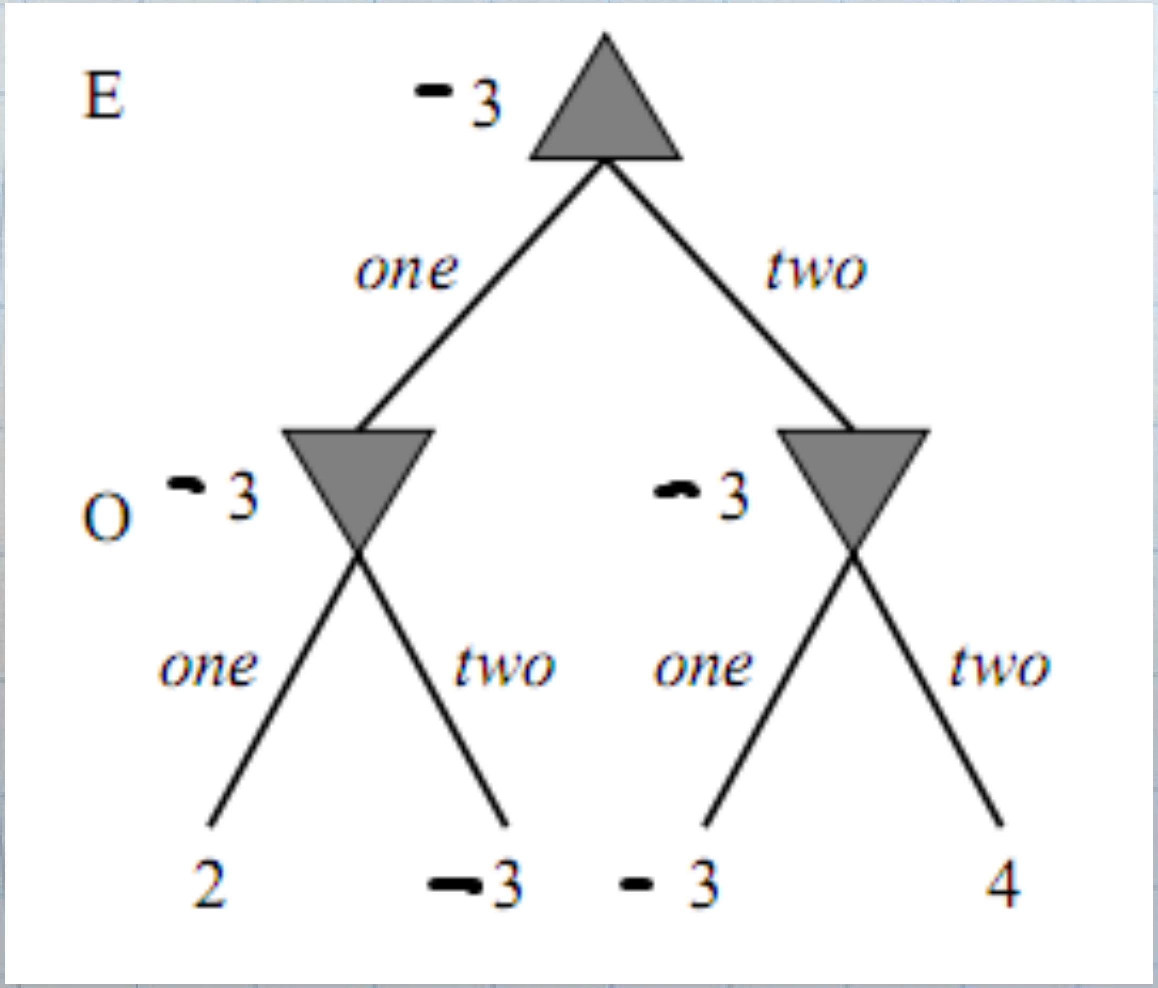
- So, no pure strategy can be an equilibrium
- And, every game has one, so it must be a mixed strategy

Finding the mixed strategy

- The maximin technique
  - For 2 player, zero-sum games
  - Finds the optimal mixed strategy
  - As in Ch 5, we choose a player to be maximizer
- For two-finger Morra
  - We choose E to be the maximizer
  - We force E to choose first, then O
  - We evaluate the expected payoffs based on strategy choices
  - To which we apply minimax algorithm
  - Then, we force O to choose before E
  - Reapply minimax

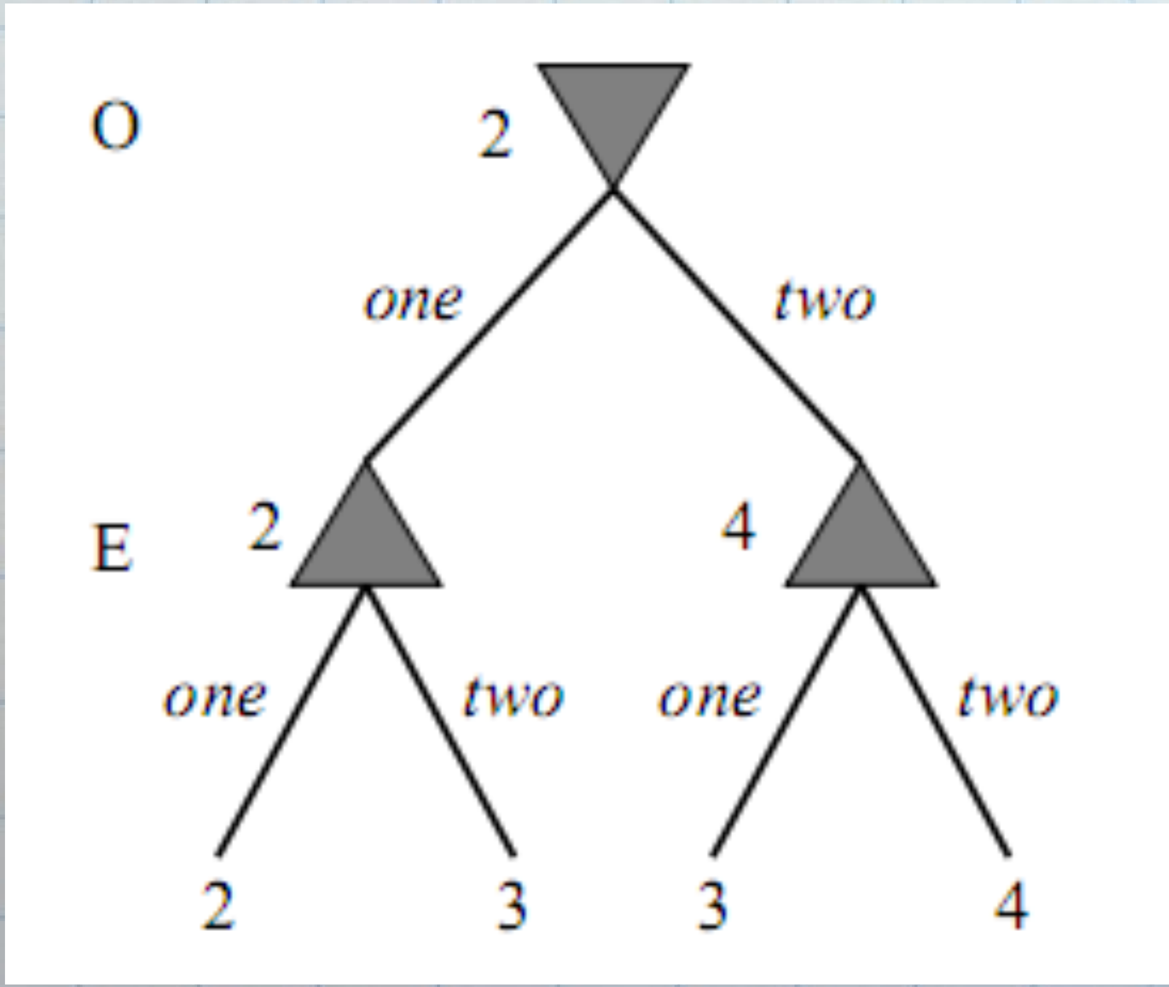
E choosing before O

The minimax tree has a root of -3, so U >= -3



O choosing before E

The root is +2, so U <= +2

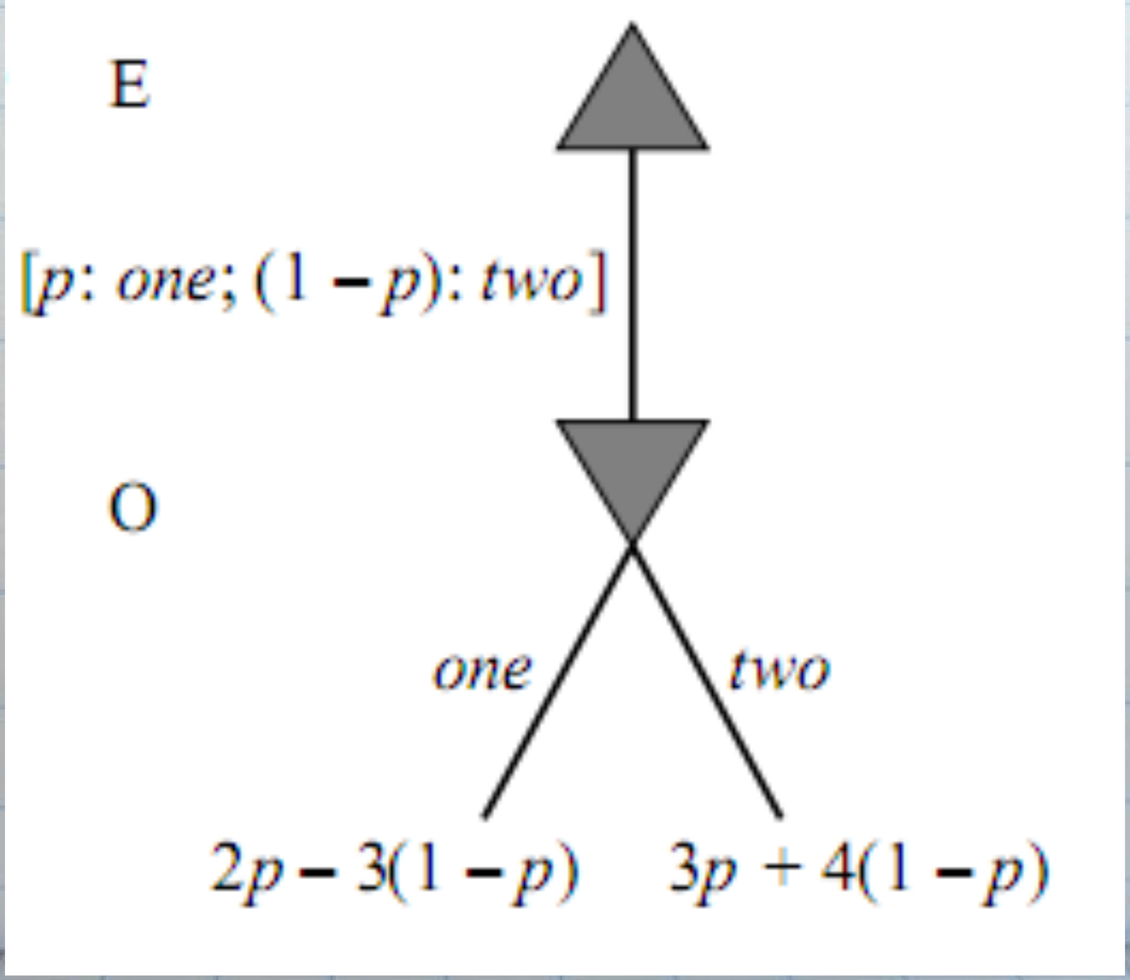


Pinpointing U

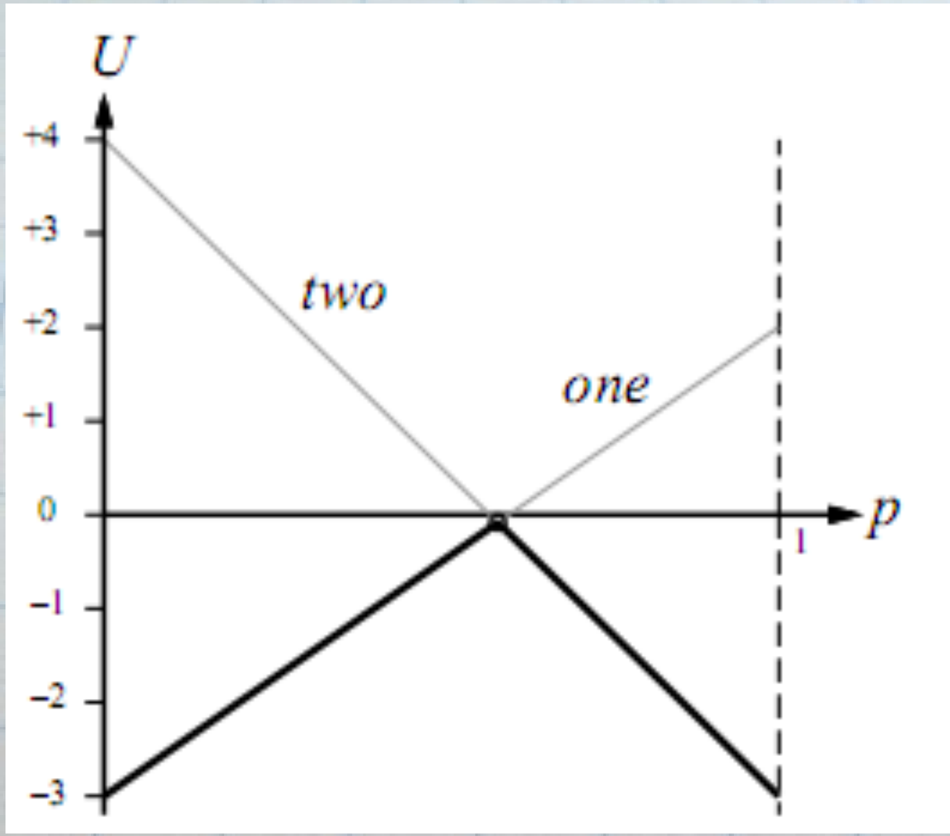
- We know that -3 <= U <= 2
- We must observe that
  - Once a player reveals their strategy, the second player might as well choose a pure strategy
  - Because, if they play mixed [p: one; (1 - p): two], the utility is a linear combo of the utilities of one and two
  - This combo can never be better than the max of one and two
- So, we can collapse the root into a single node with outward connections to player 2's pure strategy choices

If E chooses first

If O chooses one, the payoff to E = 5p - 3
If O chooses two, the payoff to E = -7p + 4



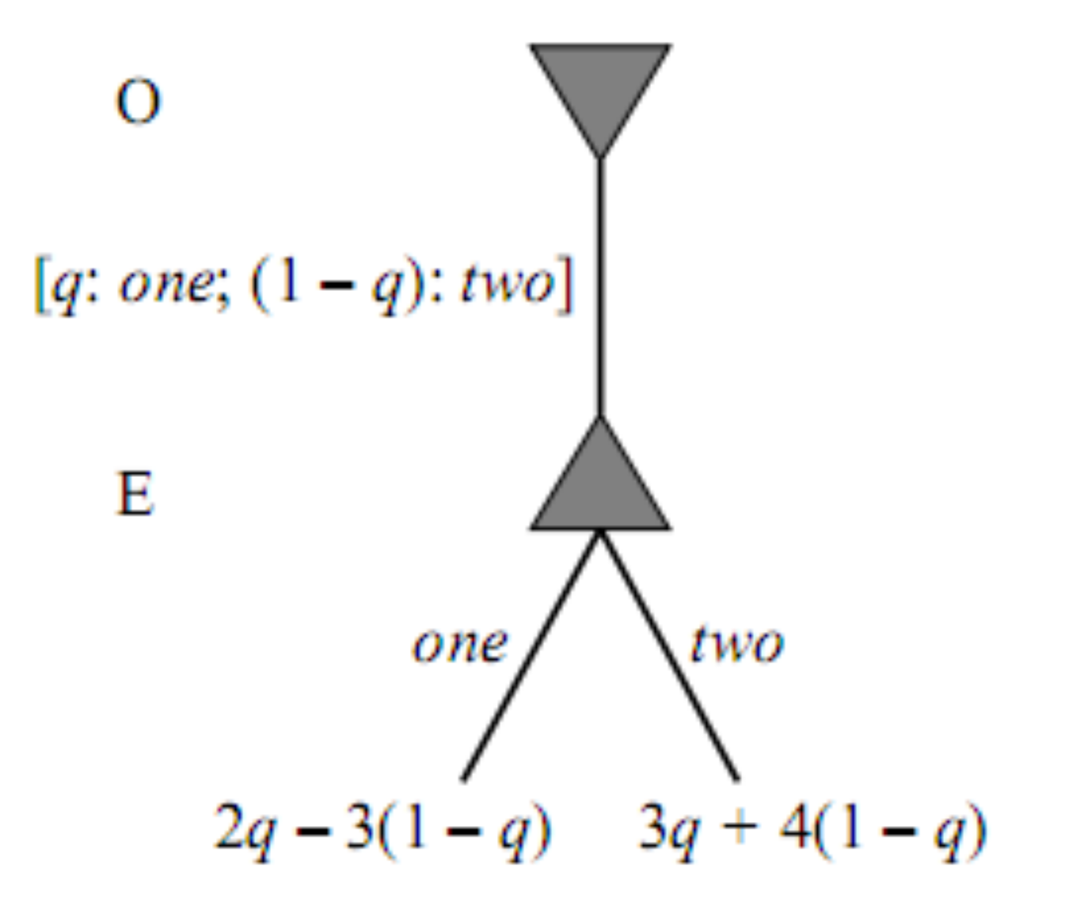
Plot payoffs for p from 0 to 1



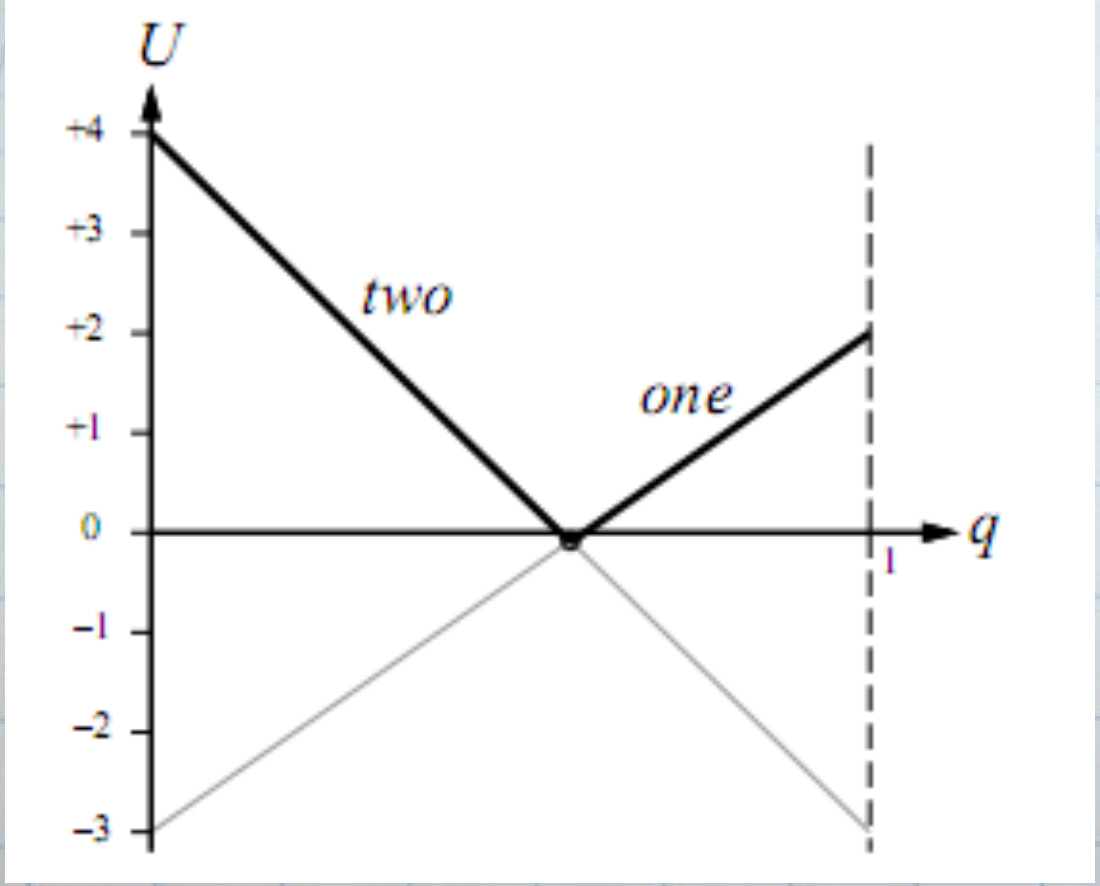
- O, the minimizer, always chooses the lower (bolded) of the two lines
- So, the best E can do is choose a p at the intersection
  - Where 5p - 3 = -7p + 4
  - p = 7/12
- And the utility for E = -1/12

If O chooses first

If E chooses one, the payoff = 5q - 3
If E chooses two, the payoff = -7q + 4



Plot for q = 0 to 1



- Again, the best O can do is at the intersection
- Intersection is where q = 7/12
- And so, utility for E = -1/12

Maximin equilibrium

- So the utility of two-finger Morra is between -1/12 and -1/12
  - i.e. It is -1/12
- Also, the mixed strategy is
  - [7/12: one; 5/12: two]
  - This is called the maximin equilibrium
  - It is also a Nash equilibrium
  - Note each component in our equilibrium mixed strategy has the same expected utility as the strategy (-1/12)

Generalized

- This result is an example of von Neumann's general result
  - "*every two player zero-sum game has a maximin equilibrium when you allow mixed strategies*"
- Also, each Nash equilibrium in a zero-sum game is a maximin for both players
- When a player adopts a maximin strategy they are guaranteed:
  - No other strategy can do better against a good opponent
  - Revealing the strategy has no impact on its effectiveness

Algorithm

- More involved than figures suggest
- With *n* possible actions, we yeild mixed strategies which are a point in *n*-dimensional space
- First remove pure strategies that are dominated
- Then, find the highest (or lowest) intersection point of all remaining hyperplanes

- This is an example of a linear programming problem, solvable in polynomial time.

- Different approach for non zero-sum games
  - Enumerate all mixed strategies (exponential in *n*)
  - Check all enumerated strategies for equilibrium

Repeated games

- Reconsider the prisoner's dilemma as a repeated game
- If Alice and Bob must play 100 rounds
  - They know the 100th round has no effect on future
    - Choose dominant *testify*
  - So the 100th is determined, now the 99th has no effect on the future
    - Also choose dominant *testify*
  - By induction, they choose (*testify* , *testify*) 100 times
    - And get 500 years in prison!
- But, if we say after each round there is a 99% chance they will meet again, more cooperation is possible

Perpetual punishment/Tit-for-tat

- If each player chooses *refuse* unless the other has chosen *testify*
  - Called perpetual punishment
  - There is no incentive to deviate from (*refuse* , *refuse*)
  - Doing so causes both to suffer a great deal
  - Works as a deterrent only if the other player believes you have adopted it
- A tit-for-tat approach is more forgiving
  - Players start with refuse and then repeat the other player's previous move
- We could also change the agents so they have no concept of the remaining steps and cannot perform induction

Sequential games

- Games with sequences of turns that may be different are best represented in tree form
  - Also known as extensive form
- The tree includes
  - Initial state S_0
  - Player(s) returns which player has the move
  - Actions(s) returns possible actions
  - Result(s, a) defines transition from s with action a
  - Utility(s, p) defines utility at terminal states
- The tree represents belief states
  - Also known as information sets
- Therefore, we can find equilibrium in the same way as normal form games

Extensive form to normal form

- Populate the normal form matrix with pure strategies
- With n information sets and a actions per set, the player will have a^n pure strategies
- So, this will only work for small games
- Workarounds?
  - Sequence form represents extensive games in linear of the size of the tree, rather than exponential
    - Still linear programming, polynomial time
  - Abstraction of game tree (removing "redundancy")

Fallbacks?

- Can deal with partially observable, multiagent, stochastic, sequential, dynamic environments
- But, does not deal well with:
  - Continuous states and actions
  - Partially defined actions
  - Less than rational opponents
  - Partially observable chance nodes
  - Partially observable utilities

Mechanism design

- Inverse game theory!
- Examples:
  - Auctioning airline tickets
  - Routing internet traffic
  - Assigning employees to work stations
  - Robotic soccer agent cooperation
  - Broadband frequency auctions

- A mechanism is
  - A description of the allowable strategies for agents
  - A center agent that collects agents' choices
  - An outcome rule (payoffs to each agent)

Auctions

- Each bidder *i* has a utility value v_i for having the item
- Sometimes v_i has a private value, sometimes common value
- Each bidder places a bid b_i
  - The bidder with b_{max} wins the item
  - Price paid need not be b_{max}

- English auction
  - Center starts asking for a minimum bid
  - If a bidder is willing to pay the minimum, center asks for b_{min} + d for some interval d.
  - If nobody is willing to pay $b_{current}$ + d, the bidder of $b_{current}$ wins the item

Is this a good mechanism?

- Need to define "good"
- Some options:
  - Maximize expected revenue for item seller
  - Maximize global utility
  - Efficient
    - Auction mechanism is efficient if the goods go to the agent with the highest v_i
  - Discourage collusion

- Example: Germany cellular spectrum auction
  - 10 blocks available
  - Each bid must be 10% more than previous bid
  - Only two serious bidders

Alternatives to English

- In general, the more bidders the better
  - Seller benefits
  - Global utility benefits
- Desirable to allow bidders to play a dominant strategy
  - Mechanism is called strategy-proof
- If the dominant strategy involves revealing the bidder's true v_i
  - This is called a truth-revealing or truthful auction
  - Any mechanism can be transformed into an equivalent truth-revealing mechanism (revelation principle)
- English auction has most of these properties
  - Simple dominant strategy
  - Almost truth-revealing
  - But, high communication cost

The sealed-bid auction

- In a sealed-bid auction
  - Each bidder sends their bid to the center
  - No other bidders see it
  - Highest bid wins the item
- No dominant strategy
  - Bid depends on estimation of other agents' bids
- Bidder with highest v_i may lose
- But, more competitive, less bias to advantaged bidders (higher resources)

The sealed-bid second-price auction

- Also known as a Vickrey auction (William Vickrey)
- A simple change
  - The winner now pays the price of the second highest bid
- Now, we have a dominant strategy
  - Simply bid v_i!

- Used broadly due to simplicity
- The seller's expected value is b_0, the same as the English auction (as d approaches 0).

Common goods

- The tragedy of the commons
- Example: air pollution
  - Each country has to decide to clean air pollution or ignore it
  - Reduce pollution cost: -10
  - Continue to pollute: -5, and -1 to all other countries
- Dominant strategy is to continue to pollute
  - 100 countries would mean -104 for each!

- So, design mechanism to avoid
  - Need to make external effects explicitly defined
  - Setting the correct price to give incentive is difficult
  - Example: a carbon tax

Distributing common goods

- We want to maximize global utility
- We can ask agent for their v_i for the item
  - They have incentive to lie
- We can use a Vickrey-Clarke-Groves (VCG) mechanism
  - The dominant strategy will be to report the true v_i
  - It works by imposing a tax equivalent to the loss in global utility that the agent is responsible for
- Algorithm:
  - The center asks each agent for their v_i
  - The center allocates goods to maximize global utility
  - The center calculates for each agent i the global utility with i in the auction and the global utility without them
  - Each agent i pays a tax equal to the difference (utility without - utility with)

Summary

- Game theory agent design
  - Can deal with partially observable, multiagent, stochastic, sequential, dynamic environments
  - Dominance, equilibrium, optimality
  - Pure/mixed strategies
  - Repeated and sequential games
  - Normal and extensive form
- Mechanism design
  - English auction
  - Strategy-proof and truth-revealing mechanisms
  - Sealed-bid auction
  - Vickrey auction
  - Tragedy of the commons
  - Vickrey-Clarke-Groves mechanism

The end!

Thanks :)