Multiagent Systems: Intro to Mechanism Design

CS 486/686: Introduction to Artificial Intelligence

Are you thinking of going to graduate school?

2nd, 3rd, and 4th year undergraduates are invited to a graduate information session.

You will get an overview of Graduate Studies, including a brief description from:

Applied Mathematics
Combinatorics & Optimization
Computational Mathematics
Computer Science
Pure Mathematics
Statistics & Actuarial Science

You will have the chance to speak with department representatives and ask questions.

Refreshments will be served.

Wednesday, November 8, 2017
DC 1301 (The "fishbowl")
4:30 - 6pm

Introduction

- So far almost everything we have looked at has been in a single-agent setting
  - Today - Multiagent Decision Making!
- For participants to act optimally, they must account for how others are going to act
- We want to
  - Understand the ways in which agents interact and behave
  - **Design systems so that agents behave the way we would like them to**

Hint for the final exam: MAS is my main research area. I like MAS problems. I even enjoy marking MAS questions. Two of the TAs for this course do MAS research. They also like marking MAS questions. There ***will*** be an MAS question on the final exam.

Mechanism Design

- Game Theory asks
  - Given a game, what should rational agents do?
- Mechanism Design asks
  - Given rational agents, *what sort of games should we design?*
  - Can we guarantee that agents will reach an outcome with properties **we** want

Fundamentals

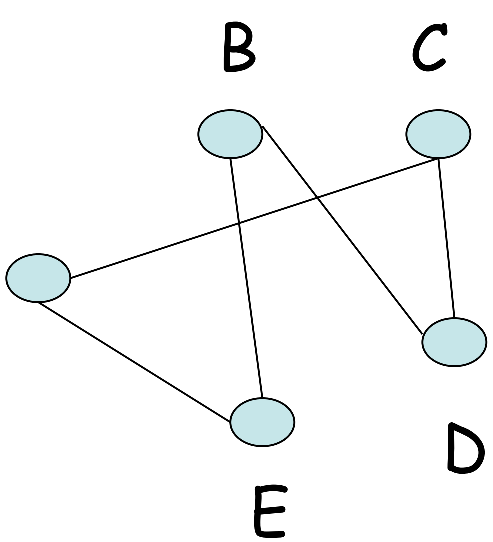
- Set of possible outcomes: O
- Set of agents: N, |N|=n
  - Each agent has a type θ_i from $\boldsymbol{\theta_i}$
  - The type captures all private information relevant to the agent's decision making
- Utility functions: $u_i(o, \theta_i)$
- Social choice function: f: $\boldsymbol{\theta_1} X \ldots X\ \boldsymbol{\theta_n} \rightarrow O$

Examples of Social Choice Functions

- **Voting**
  - Choose a candidate from amongst a group
- **Public Project**
  - Decide whether to build a road whose cost must be funded by the agents themselves
- **Allocation**
  - Allocate an item or resource to one agent in the group

Scenario

- Network routing problem to allocate resources to minimize the total cost of delay over all agents

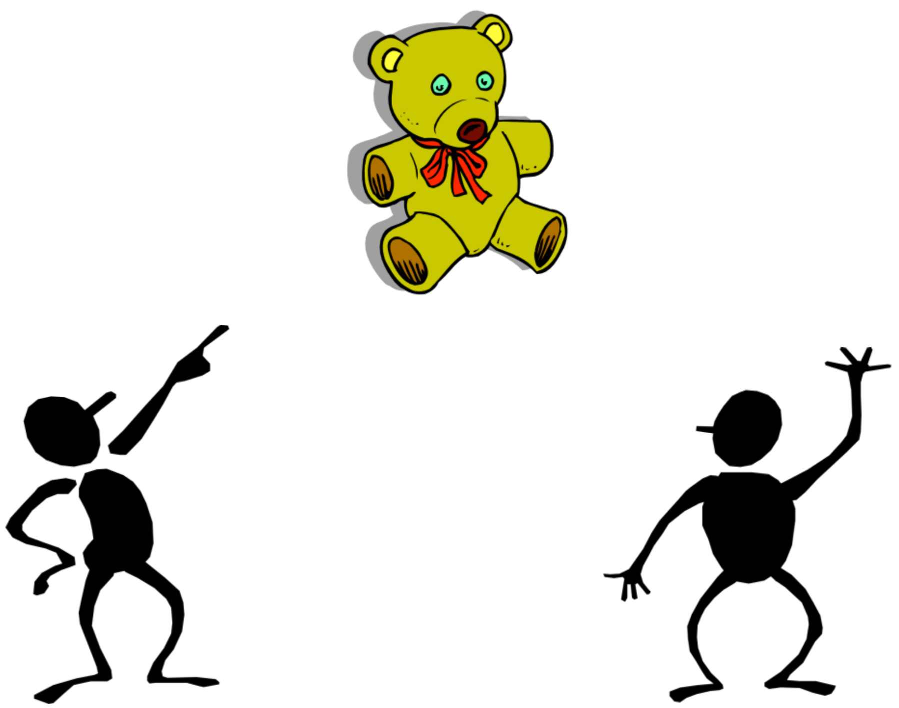


My unit cost of delay for sending messages from A to D is $1

My unit cost of delay for sending messages between E and D is $5

A Potential Problem

- Agents’ types are not public, and agents are acting in their own self-interest



Mechanism Design Problem

- By having agents interact through an "institution" we might be able to solve this problem
- **Mechanism**

$$M = (S_1, \ldots, S_n, g(\cdot))$$

  - S_i is the strategy space of agent i
  - $g: S_1 \times \ldots \times S_n \rightarrow O$ is the outcome function

Implementation

- A mechanism $M=(S_1,\ldots,S_n,g())$ **implements** social choice function $f(\theta)$ if there is an equilibrium s^*

$$s^* = (s_1^*(\theta_1), \ldots, s_n^*(\theta_n))$$

such that

$$g(s_1^*(\theta_1), \ldots, s_n^*(\theta_n)) = f(\theta_1, \ldots, \theta_n)$$

for all $(\theta_1, \ldots, \theta_n) \in \Theta_1 \times \Theta_n$

Direct Mechanisms

- A direct mechanism is a mechanism where

$$S_i = \Theta_i \text{ for all } i$$

and

$$g(\theta) = f(\theta) \text{ for all } \theta \in \Theta_1 \times \Theta_n$$

Incentive Compatibility

- A direct mechanism is **incentive compatible** if it has an equilibrium s* where

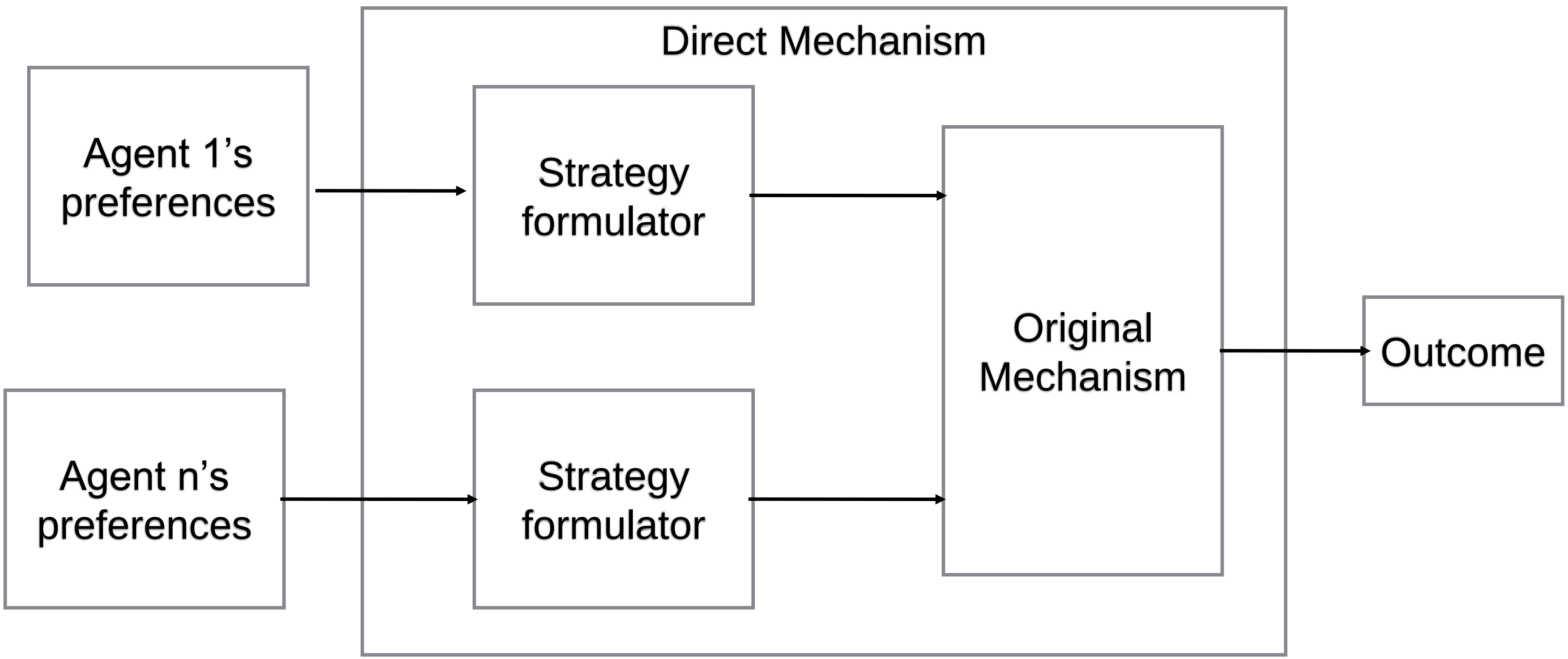
$$s_i^*(\theta_i) = \theta_i$$

  for all θ_i in Θ_i and for all i.

  - A direct mechanism is strategy proof if the equilibrium above is a dominant strategy equilibrium

Revelation Principle

- **Theorem**: Suppose there exists a mechanism M that implements social choice function f in dominant strategies. Then there is a direct strategy-proof mechanism M' which *also implements* f.



Quick Review

- We know
  - What a mechanism is
  - What it means for a SCF to be (dominant-strategy) implementable
  - Revelation Principle
- We do not yet know
  - What types of SCF are dominant-strategy implementable

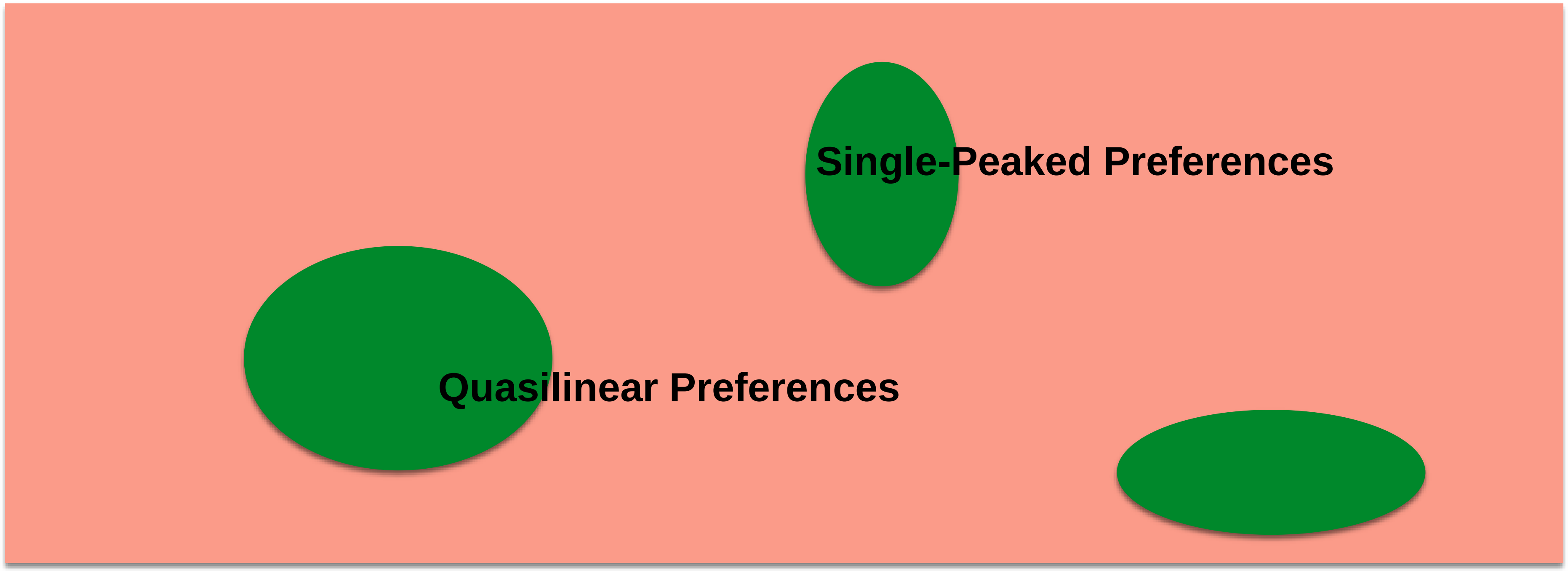
Gibbard-Satterthwaite Theorem

- **Theorem**: Assume that
  - O is finite and $|O|>2$
  - Each o in O can be achieved by SCF f for some θ
  - θ includes all possible strict orderings over O

Then f is implementable in dominant strategies if and only if f is *dictatorial*.

Circumventing Gibbard-Satterthwaite

- Use a weaker equilibrium concept
- Design mechanisms where computing manipulations is computationally hard
- Restrict the structure of agents' preferences



Single-Peaked Preferences

- Median-Voter rule is strategy proof for single-peaked preferences

Quasilinear Preferences

- Outcome $o=(x,t_1,\ldots,t_n)$
  - x is a “project choice”
  - t_i in $\mathbb{R}$ are transfers (“money”)
- Utility functions: $u_i(o,\theta_i)=v_i(x,\theta_i)-t_i$
- Quasilinear mechanism $M=(S_1,\ldots,S_n,g())$ where
  - $g()=(x(),t_1,\ldots,t_n)$

Groves Mechanisms

- Choice rule

$$x^*(\theta) = \arg\max_x \sum_i v_i(x, \theta_i)$$

- Transfer rules

$$t_i(\theta) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(x^*(\theta), \theta_j)$$

Groves Mechanisms

- **Theorem**: Groves mechanisms are strategy-proof and efficient.

- **Theorem**: Groves mechanisms are unique (up to $h_i(\theta_{-i})$)

Vickrey-Clarke-Groves Mechanism

- Outcome

$$x^* = \arg\max_x \sum_i v_i(x, \theta_i)$$

- Transfers

$$t_i(\theta) = \sum_{j \neq i} v_j(x^{-i}, \theta_j) - \sum_{j \neq i} v_j(x^*(\theta), \theta_j)$$

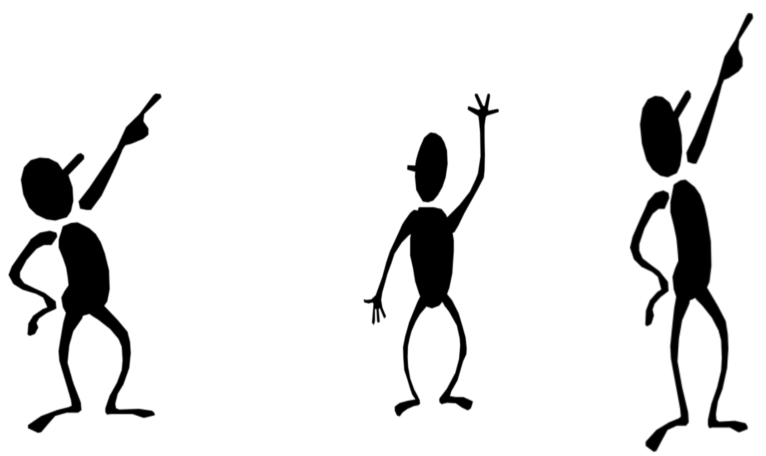
- VCG is an example of a Groves mechanism
  - Efficient and strategy-proof
  - Agents' equilibrium utility is their marginal contribution to the welfare of the system

Example: Allocation Problem

- Social choice function
  - Maximize social welfare (i.e. give item to agent who values it the most)
- Utility functions: $u_i = v_i(o) - t_i$
- Mechanism (Vickrey Auction)
  - S_i: a bid of any non-negative number
  - Outcome function g:
    - Give item to agent who submits highest bid
    - Highest bidder pays amount of second highest bid, all else pay nothing

Vickrey Auction

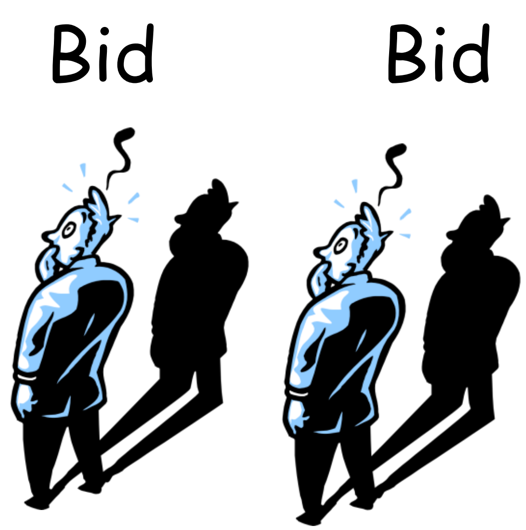
$V_1=\$6$ $V_2=\$5$ $V_3=\$2$



Another Application: Sponsored Search

Slot 1
Slot 2
Slot 3
Slot 4
Slot 5

<Keyword>



1. Advertisers are ranked and assigned slots based on the ranking.
2. If an ad is clicked on, only then does the advertiser pay.

Ranking

- Rank-by-relevance
  - Assign slots of order of (quality score)*(bid)

Bidder	Bid	Quality Score
A	1.50	0.5
B	1.00	0.9
C	0.75	1.5

→

Ranking
C (1.25)
B (0.9)
A (0.75)

Pricing

- An advertiser only pays when its ad is clicked on
- How much does it pay?
  - The *lowest price* it could have bid and still been in the same position

Example

Bidder	Bid	Quality Score
A	1.50	0.5
B	1.00	0.9
C	0.75	1.5

Ranking
C (1.25)
B (0.9)
A (0.75)

C will pay p=0.9/1.5=0.6
B will pay p=0.75/0.9 = 0.83

How much will A pay?

Sponsored Search

- How would you design a bidding agent for sponsored search?
- Different from the Vickrey auction
  - There is no single best strategy
  - *It depends on the strategies of others*

Summary

- Definition of a mechanism
- What it means for a mechanism to implement a social choice function
- Revelation Principle
- Gibbard-Satterthwaite Theorem
- Possibility results
  - Groves mechanisms