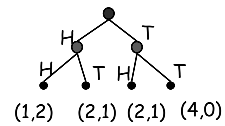
Mechanism Design

CS 886
Electronic Market Design
University of Waterloo

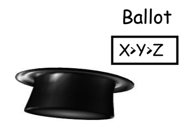
Introduction

So far we have looked at

- Game Theory
  - Given a game we are able to analyze the strategies agents will follow



- Social Choice Theory
  - Given a set of agents' preferences we can choose some outcome



Introduction

- Today, Mechanism Design
  - Game Theory + Social Choice
- Goal of Mechanism Design is to
  - Obtain some outcome (function of agents' preferences)
  - But agents are rational
    - They may lie about their preferences
- Goal: Define the rules of a game so that in equilibrium the agents do what we want

Fundamentals

- Set of possible outcomes, *O*
- Agents $i \in I$, $|I|=n$, each agent i has type $\theta i \in \Theta i$
  - Type captures all private information that is relevant to agent's decision making
- Utility $ui(o, \theta i)$, over outcome $o \in O$
- Recall: goal is to implement some system-wide solution
  - Captured by a social choice function

$$f: \Theta_1 \times \dots \times \Theta_n \rightarrow \mathcal{O}$$

$f(\theta_1, \dots \theta_n)=o$ is a collective choice

Examples of social choice functions

- Voting: choose a candidate among a group
- Public project: decide whether to build a swimming pool whose cost must be funded by the agents themselves
- Allocation: allocate a single, indivisible item to one agent in a group

Mechanisms

- Recall: We want to implement a social choice function
  - Need to know agents' preferences
  - They may not reveal them to us truthfully
- Example:
  - 1 item to allocate, and want to give it to the agent who values it the most
  - If we just ask agents to tell us their preferences, they may lie

Mechanism Design Problem

- By having agents interact through an institution we might be able to solve the problem
- Mechanism:

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

Strategy spaces of agents — Outcome function

$$g: S_1 \times \ldots \times S_n \rightarrow O$$

Implementation

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

implements social choice function $f(\theta)$

if there is an equilibrium strategy profile $s^*(\cdot)=(s^*_1(\cdot),\ldots,s^*_n(\cdot))$

of the game induced by M such that

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

for all

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

Implementation

- We did not specify the type of equilibrium in the definition
- Nash

$$u_i(s_i^*(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i) \geq u_i(s_i'(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i),\ \forall i,\ \forall \theta,\ \forall s_i' \neq s_i^*$$

- Bayes-Nash

$$E[u_i(s_i^*(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i)] \geq E[u_i(s_i'(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i)],\ \forall i,\ \forall \theta,\ \forall s_i' \neq s_i^*$$

- Dominant

$$u_i(s_i^*(\theta_i),s_{-i}(\theta_i),\theta_i) \geq u_i(s_i'(\theta_i),s_{-i}(\theta_{-i}),\theta_i),\ \forall i,\ \forall \theta,\ \forall s_i' \neq s_i^*,\ \forall s_{-i}$$

Direct Mechanisms

- Recall that a mechanism specifies the strategy sets of the agents
  - *These sets can contain complex strategies*
- **Direct mechanisms:**
  - Mechanism in which $S_i=\Theta_i$ for all i, and $g(\theta)=f(\theta)$ for all $\theta \in \Theta_1 \times \ldots \times \Theta_n$
- **Incentive compatible:**
  - A direct mechanism is incentive compatible if it has an equilibrium s^* where $s^*_i(\theta_i)=\theta_i$ for all $\theta_i \in \Theta_i$ and all i
  - (truth telling by all agents is an equilibrium)
  - Strategy-proof if dominant-strategy equilibrium

Dominant Strategy Implementation

- Is a certain social choice function implementable in dominant strategies?
  - In principle we would need to consider all possible mechanisms
- **Revelation Principle** (for Dom Strategies)
  - Suppose there exists a mechanism $M=(S_1,\ldots,S_n,g(\cdot))$ that implements social choice function f() in dominant strategies. Then there is a direct strategy-proof mechanism, M', which also implements f().

Revelation Principle

"the computations that go on within the mind of any bidder in the nondirect mechanism are shifted to become part of the mechanism in the direct mechanism" [McAfee&McMillian 87]

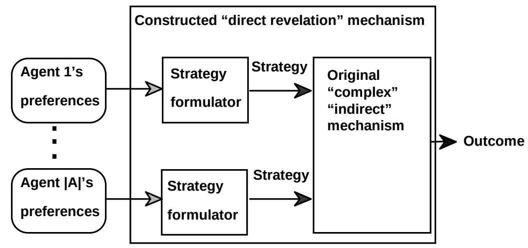
- Consider the incentive-compatible direct-revelation implementation of an English auction

Revelation Principle: Proof

- $M=(S_1,\ldots,S_n,g())$ implements SCF f() in dom str.
  - Construct direct mechanism $M'=(\Theta^n,f(\theta))$
  - By contradiction, assume
    $\exists\ \theta_i'\neq\theta_i$ s.t. $u_i(f(\theta_i',\theta_{-i}),\theta_i) \succ u_i(f(\theta_i,\theta_{-i}),\theta_i)$
    for some $\theta_i'\neq\theta_i$, some θ_{-i}.
  - But, because $f(\theta)=g(s^*(\theta))$, this implies
    $u_i(g(s_i^*(\theta_i'),s_{-i}^*(\theta_{-i})),\theta_i) \succ u_i(g(s^*(\theta_i),s^*(\theta_{-i})),\theta_i)$

    Which contradicts the strategy proofness of s^* in M

Revelation Principle: Intuition



Theoretical Implications

- Literal interpretation: Need only study direct mechanisms
    - This is a smaller space of mechanisms
  - Negative results: If no direct mechanism can implement SCF f() then no mechanism can do it
  - Analysis tool:
    - Best direct mechanism gives us an upper bound on what we can achieve with an indirect mechanism
    - Analyze all direct mechanisms and choose the best one

Practical Implications

- Incentive-compatibility is "free" from an implementation perspective
- **BUT!!!**
  - A lot of mechanisms used in practice are not direct and incentive-compatible
  - Maybe there are some issues that are being ignored here

Quick review

- We now know
  - What a mechanism is
  - What is means for a SCF to be dominant strategy implementable
  - If a SCF is implementable in dominant strategies then it can be implemented by a direct incentive-compatible mechanism
- We do not know
  - What types of SCF are dominant strategy implementable

Gibbard-Satterthwaite Thm

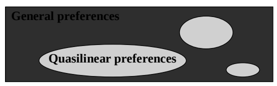
- Assume
  - $\mathcal{O}$ is finite and $|\mathcal{O}|\geq 3$
  - Each $o\in\mathcal{O}$ can be achieved by social choice function f() for some θ

Then:

f() is truthfully implementable in dominant strategies if and only if f() is dictatorial

Circumventing G-S

- Use a weaker equilibrium concept
  - Nash, Bayes-Nash
- Design mechanisms where computing a beneficial manipulation is hard
  - Many voting mechanisms are NP-hard to manipulate (or can be made NP-hard with small "tweaks) [Bartholdi, Tovey, Trick 89] [Conitzer, Sandholm 03]
- Randomization  Almost need this much
- Agents' preferences have special structure



Quasi-Linear Preferences

- Outcome $o=(x,t_1,\dots,t_n)$
  - x is a "project choice" and $t_i \in \mathbb{R}$ are transfers (money)
- Utility function of agent i
  - $u_i(o,\theta_i)=u_i((x,t_1,\dots,t_n),\theta_i)=v_i(x,\theta_i)-t_i$
- Quasi-linear mechanism: $M=(S_1,\dots,S_n,g(\cdot))$ where $g(\cdot)=(x(\cdot),t_1(\cdot),\dots,t_n(\cdot))$

Social choice functions and quasi-linear settings

- SCF is efficient if for all types $\theta=(\theta_1,\dots,\theta_n)$
  - $\sum_{i=1}^n v_i(x(\theta),\theta_i) \geq \sum_{i=1}^n v_i(x'(\theta),\theta_i) \quad \forall\ x'(\theta)$
  - Aka social welfare maximizing
- SCF is budget-balanced if
  - $\sum_{i=1}^n t_i(\theta)=0$
  - Weakly budget-balanced if $\sum_{i=1}^n t_i(\theta) \geq 0$

Groves Mechanisms

[Groves 1973]

- A **Groves mechanism**, $M=(S_1,\dots,S_n,(x,t_1,\dots,t_n))$ is defined by
  - <u>Choice rule</u> $x^*(\theta')=\arg\max_x \sum_i v_i(x,\theta_i')$
  - <u>Transfer rules</u>
    - $t_i(\theta')=h_i(\theta_{-i}')-\sum_{j\neq i} v_j(x^*(\theta'),\theta'_j)$

  where $h_i(\cdot)$ is an (arbitrary) function that does not depend on the reported type θ_i' of agent i

Groves Mechanisms

- **Thm:** Groves mechanisms are strategy-proof and efficient (We have gotten around Gibbard-Satterthwaite!)
- Proof: Agent i's utility for strategy θ_i', given θ_{-i}' from agents $j\neq i$ is

  $U_i(\theta_i')=v_i(x^*(\theta'),\theta_i)-t_i(\theta')$

  $=v_i(x^*(\theta'),\theta_i)+\sum_{j\neq i} v_j(x^*(\theta'),\theta'_j)-h_i(\theta'_{-i})$

  Ignore $h_i(\theta_{-i})$. Notice that

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

  i.e. it maximizes the sum of reported values.

  Therefore, agent i should announce $\theta_i'=\theta_i$ to maximize its own payoff

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

VCG Mechanism

(aka Clarke mechanism aka Pivotal mechanism)

- Def: Implement efficient outcome,

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

  Compute transfers

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

  Where $x^{-i}=\max_x \sum_{j\neq i} v_j(x,\theta_j')$

VCG are efficient and strategy-proof

Agent's equilibrium utility is:

$u_i(x^*,t_i,\theta_i)=v_i(x^*,\theta_i)-[\sum_{j\neq i} v_j(x^{-i},\theta_j) - \sum_{j\neq i} v_j(x^*,\theta_j)]$

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

= marginal contribution to the welfare of the system

Example: Building a pool

- The cost of building the pool is $300
- If together all agents value the pool more than $300 then it will be built
- Clarke Mechanism:
  - Each agent announces their value, v_i
  - If $\sum v_i \geq 300$ then it is built
  - Payments $t_i(\theta_i') = \sum_{j\neq i} v_j(x^{-i}, \theta_j') - \sum_{j\neq i} v_j(x^*, \theta_i')$ if built, 0 otherwise

v1=50, v2=50, v3=250

Pool should be built

$t_1=(250+50)-(250+50)=0$
$t_2=(250+50)-(250+50)=0$
$t_3=(0)-(100)=-100$

Not budget balanced

Vickrey Auction

- Highest bidder gets item, and pays second highest amount
- Also a VCG mechanism
  - Allocation rule: get item if $b_i=\max_i[b_j]$
  - Every agent pays

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

$\max_{j\neq i}[b_j]$

$\max_{j\neq i}[b_j]$ if i is not the highest bidder,

0 if it is

London Bus System

(as of April 2004)

- 5 million passengers each day
- 7500 buses
- 700 routes

- The system has been privatized since 1997 by using competitive tendering
- Idea: Run an auction to allocate routes to companies

The Generalized Vickrey Auction

(VCG mechanism)

- Let *G* be set of all routes, *I* be set of bidders
- Agent *i* submits bids $v_i^*(S)$ for all bundles $S \subseteq G$
- Compute allocation S* to maximize sum of reported bids

$$V^*(I)=\max_{(S1,\ldots,SI)}\sum_i v_i^*(S_i)$$

- Compute best allocation without each agent *i*:

$$V^*(I\setminus i)=\max_{(S1,\ldots,SI)}\sum_{j\neq i} v_i^*(S_i)$$

- Allocate each agent Si*, each agent pays

$$P(i)=v_i^*(S_i^*)-[V^*(I)-V^*(I\setminus i)]$$