Mechanism Design

1: Definitions, Implementation

Egor Starkov

Københavns Unversitet
Fall 2024

This slide deck:

1. Defining a Mechanism
2. Revelation Principle
3. Dominant Strategy Implementation
4. Bayesian Implementation

What is a mechanism?

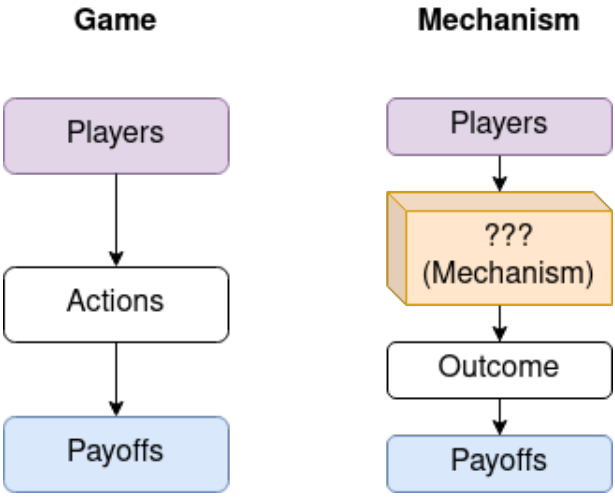
Let's reverse engineer from a simpler question: **What is a game?**

1. Set of players $i \in \{1, ..., N\}$
2. Set of actions A_i for every i; set of action profiles $A \equiv \times_{i \in N} A_i$
3. Collection of utility functions $u_i : A \to \mathbb{R}$

(This is a *normal-form game*. All extensive-form games ("trees") and incomplete-information games can be represented as normal-form games.)

Which parts of this definition are fixed at a higher level, and which can we *design* as a part of a *mechanism*?

Problem environment



General Problem Set-up

In our MD problem, the following environment will be **fixed**:

- N agents,
- set X of outcomes,
- each agent i has type $\theta_i \in \Theta_i$:
  - describes agent's information,
  - describes agent's preferences;
- the type profile $\theta = (\theta_1, \ldots, \theta_N)$ is distributed according to a distribution F with p.d.f. ϕ,
  - (often a missing subscript denotes a vector of respective objects)
  - distribution F is commonly known and agreed upon
- each agent has a utility function $u_i(x, \theta_i)$ that depends on the collective choice $x \in X$ and his type θ_i,

Mechanism

- a mechanism is a game played by the agents
- each agent has an action set A_i in this game

Definition (mechanism)

A mechanism $\Gamma = (A_1, \ldots, A_N, g(\cdot))$ is a collection of:

- N strategy sets $(A_1, \ldots, A_N)$ and
- an outcome function $g : A_1 \times \cdots \times A_N \to X$.

Social Choice Function

Definition (Social choice function)

A social choice function is a function $f : \Theta_1 \times \cdots \times \Theta_N \to X$ that assigns to each profile of types $(\theta_1, \ldots, \theta_N)$ a collective choice $f(\theta_1, \ldots, \theta_N) \in X$.

- gives a desired outcome as a function of the agents' types

Implementation

Definition (implementation)

Mechanism $\Gamma = (A_1, \ldots, A_N, g(\cdot))$ implements the s.c.f. f if there is an equilibrium strategy profile $(a_1^*, \ldots, a_N^*)$ of the Bayesian game induced by Γ such that

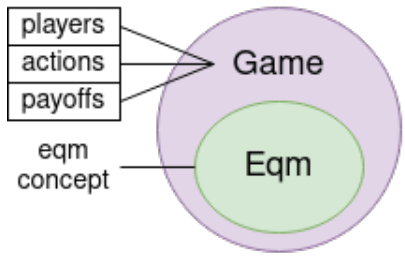
$$g(a_1^*(\theta_1), \ldots, a_N^*(\theta_N)) = f(\theta_1, \ldots, \theta_N)$$

for all $(\theta_1, \ldots, \theta_N) \in \Theta_1 \times \cdots \times \Theta_N$.

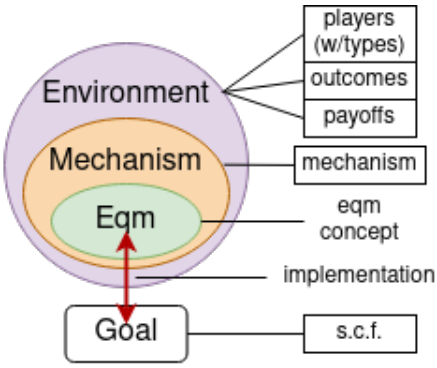
Summary of definitions

- S.c.f. f describes what we want to achieve;
- Mechanism $\Gamma = (S, g)$ describes what we do and how;
- Implementability says whether we have achieved our goal.

Full problem setup



Game Theory problem



Mechanism Design problem

This slide deck:

Revelation Principle

- Main cheat in Mechanism Design! No need to bruteforce through uncountable numbers of different games! It is enough to just... (click to see more)

Revelation Principle

- Main cheat in Mechanism Design! No need to bruteforce through uncountable numbers of different games! It is enough to just...
- Instead of making players play the game, ask them for their θ_i and promise to play on their behalf!
- Requires that the designer has commitment power.
  - Strong assumption, sometimes reasonable(?)
  - The necessary evil for our purposes.
  - Useful for formal analysis, but in the end the resulting mechanism can *sometimes* work even without commitment.

Revelation Principle: Definitions

Fix some s.c.f. $f : \Theta \to X$.

Definition (Direct revelation mechanism)

A direct revelation mechanism for f is a mechanism in which $A_i = \Theta_i$ for all i and $g(\theta) = f(\theta)$

Definition (Truthful implementation)

S.c.f. f is truthfully implementable if it can be implemented by a direct revelation mechanism.

Revelation Principle: Statement

Revelation principle (blanket statement)

Suppose there exists a mechanism $\Gamma = (A_1, \ldots, A_N, g)$ that implements the social choice function f.
Then f is truthfully implementable.

- The "theorem" above is informal.
  - "Implementation" requires "an equilibrium", which can mean a million different things.
  - We will now plug in some specific equilibrium concepts.

This slide deck:

Game Theory Recap: Dominant Strategy

- strategy a_i is a full contingent plan of play
- strategy a_i is dominant for agent i if it is best *no matter what the other players do*

Definition (dominant strategy)

Given mechanism $\Gamma = (A, g)$, $a_i : \Theta_i \to A_i$ is a dominant strategy if for all $\theta_i \in \Theta_i$

$$u_i(g(a_i(\theta_i), a_{-i}), \theta_i) \geq u_i(g(\hat{a}_i, a_{-i}), \theta_i)$$

for all $\hat{a}_i \in A_i$ and all $a_{-i} \in A_{-i}$.

- our definition slightly different from the standard – does not require strict inequality

Dominant Strategy Equilibrium

- in a dominant strategy equilibrium every player plays a dominant strategy

Definition (dominant strategy equilibrium)

A strategy profile $(a_1^*, \ldots, a_N^*)$ is a dominant strategy equilibrium of mechanism $\Gamma = (A_1, \ldots, A_N, g)$ if for all i and all $\theta_i \in \Theta_i$

$$u_i(g(a_i^*(\theta_i), a_{-i}), \theta_i) \geq u_i(g(\hat{a}_i, a_{-i}), \theta_i)$$

for all $\hat{a}_i \in A_i$ and all $a_{-i} \in A_{-i}$.

Now let's finally be formal about all our definitions.

Dominant Strategy Implementation

- A mechanism implements f in dominant strategies if
  - the game induced by the mechanism has a dominant strategy equilibrium
  - the outcome in this equilibrium coincides with f

Definition (implementation in dominant strategies)

A mechanism $\Gamma = (A_1, \ldots, A_N, g)$ implements the social choice function f in dominant strategies if there exists a dominant strategy equilibrium $(a_1^*, \ldots, a_N^*)$ of Γ such that $g(a_1^*(\theta_1), \ldots, a_N^*(\theta_N)) = f(\theta)$ for all $\theta \in \Theta$.

Good Implementation Concept?

- very robust equilibrium concept
  - no need to predict what the other players will play
  - no need to know the type distribution ϕ
  - works even if
    - players don't know ϕ or even if players believe in different ϕ_i (protects from players' model misspecification)
    - players think that other players are not rational
- not a panacea
  - does not rule out other weird Nash Equilibria (example: second-price auction)
  - is not necessarily collusion-proof
  - does not protect from designer's model misspecification

Bottom line: it's as good as they get, but far from perfect.

Dominant Strategy Incentive Compatibility

Theorem (Revelation Principle for Dominant Strategies)

Suppose there exists a mechanism $\Gamma = (A_1, \ldots, A_N, g)$ *that implements the social choice function* f *in dominant strategies.*
Then f *is truthfully implementable in dominant strategies.*

Definition (Dominant Strategy Incentive Compatibility)

"f is dominant strategy incentive compatible (DSIC)"
means the exact same thing as
"f is truthfully implementable in dominant strategies".

DS Revelation Principle: Proof

Let Γ implement f in dominant strategies, i.e. there is a strategy profile $(a_1^*, \ldots, a_N^*)$ such that $g(a_1^*(\theta_1), \ldots, a_N^*(\theta_N)) = f(\theta)$ for all θ, and for all i and $\theta_i \in \Theta_i$,

$$u_i(g(a_i^*(\theta_i), a_{-i}), \theta_i) \geq u_i(g(\hat{a}_i, a_{-i}), \theta_i)$$

for all $\hat{a}_i \in A_i$ and all $a_{-i} \in A_{-i}$.

Then

$$u_i(g(a_i^*(\theta_i), a_{-i}^*(\theta_{-i})), \theta_i) \geq u_i(g(a_i^*(\hat{\theta}_i), a_{-i}^*(\theta_{-i})), \theta_i)$$

for all $\hat{\theta}_i \in \Theta_i$, $\theta_{-i} \in \Theta_{-i}$.

Since $g(a^*(\theta)) = f(\theta)$,

$$u_i(f(\theta_i, \hat{\theta}_{-i}), \theta_i) \geq u_i(f(\hat{\theta}_i, \hat{\theta}_{-i}), \theta_i)$$

for all $\hat{\theta}_{-i} \in \Theta_{-i}$.

Revelation Principle: Is it cool or is it cool?

- Idea: to solve the problem mathematically, it is enough to only look at direct mechanisms!
  - This result allows to quickly check whether a given f is [DS] implementable.
  - If yes, gives you a mechanism to implement it.
  - If not, helps you describe a set of implementable s.c.f. and pick second best.
  - *Yours today for ~~only $49.99+shipping~~ FREE with a qualifying Mechanism Design course!*
- Translating that solution to the real world may (and often does) result in an indirect mechanism! We'll see some examples.

This slide deck:

DSIC vs BIC

- We've looked at DS implementation so far. Robust but demanding. Can we get more mileage by relaxing the equilibrium notion?
- Now: use standard Bayes-Nash Equilibrium as solution concept. Weaker equilibrium concept, so:
  - we are less confident it will produce the intended outcome, but
  - it can implement more(?) s.c.f-ns.
  - (there's a literature studying whether sets of DSIC and BIC s.c.f-ns are equal in special settings)

Bayesian Implementation

Start with the general model as before:

- N agents;
- set of alternatives X;
- type $\theta_i \in \Theta_i$ is private information of i;
- common prior belief $\phi \in \Delta(\Theta)$ about distribution of types;
- utility functions $u_i(x, \theta_i)$;
- each agent uses Bayes' rule to form a belief over other agents' types

$$\phi(\theta_{-i}|\theta_i) = \phi(\theta_i, \theta_{-i}|\theta_i) = \frac{\phi(\theta_i, \theta_{-i})}{\int_{\tilde{\theta}_{-i} \in \Theta_{-i}} \phi(\theta_i, \tilde{\theta}_{-i}) d\tilde{\theta}_{-i}}.$$

Bayes-Nash Equilibrium

Definition (Bayes-Nash equilibrium)

The strategy profile $a^* = (a_1^*, \ldots, a_N^*)$ with $a_i^* : \Theta_i \to A_i$ is a Bayes-Nash equilibrium of the mechanism $\Gamma = (A_1, \ldots, A_N, g)$ if, for all i and all $\theta_i \in \Theta_i$,

$$\mathbb{E}_{\theta_{-i}}\left[u_i(g(a_i^*(\theta_i), a_{-i}^*(\theta_{-i})), \theta_i)|\theta_i\right] \geq \mathbb{E}_{\theta_{-i}}\left[u_i(g(\hat{a}_i, a_{-i}^*(\theta_{-i})), \theta_i)|\theta_i\right]$$

for all $\hat{a}_i \in A_i$.

Bayes-Nash Equilibrium

Definition (Bayes-Nash equilibrium)

The strategy profile $a^* = (a_1^*, \ldots, a_N^*)$ with $a_i^* : \Theta_i \to A_i$ is a Bayes-Nash equilibrium of the mechanism $\Gamma = (A_1, \ldots, A_N, g)$ if, for all i and all $\theta_i \in \Theta_i$,

$$\mathbb{E}_{\theta_{-i}}\left[u_i(g(a_i^*(\theta_i), a_{-i}^*(\theta_{-i})), \theta_i)|\theta_i\right] \geq \mathbb{E}_{\theta_{-i}}\left[u_i(g(\hat{a}_i, a_{-i}^*(\theta_{-i})), \theta_i)|\theta_i\right]$$

for all $\hat{a}_i \in A_i$.

- Standard NE reasoning: if everyone else plays eqm strats, i has no incentive to deviate.
- (This definition is for pure strategies, but there is no problem in allowing mixed strategies.)
- Expectations are taken w.r.t. distribution $F(\theta)$

Bayesian Implementation

Definition (Bayesian implementation)

Mechanism $\Gamma = (A_1, \ldots, A_N, g)$ implements s.c.f. f in Bayes-Nash equilibrium if there is a BNE $a^* = (a_1^*, \ldots, a_N^*)$ of Γ such that $f(\theta) = g(a^*(\theta))$ for all $\theta \in \Theta$.

Bayesian Implementation

Definition (Bayesian implementation)

Mechanism $\Gamma = (A_1, \ldots, A_N, g)$ implements s.c.f. f in Bayes-Nash equilibrium if there is a BNE $a^* = (a_1^*, \ldots, a_N^*)$ of Γ such that $f(\theta) = g(a^*(\theta))$ for all $\theta \in \Theta$.

Definition (Bayesian implementability)

S.c.f. f is implementable in BNE if there exists Γ which implements it in BNE.

Truthful Bayesian Implementation

Definition (Truthful Bayesian implementation)

S.c.f. f is truthfully implementable in BNE (=Bayesian Incentive Compatible, BIC) if $a_i^*(\theta_i) = \theta_i$ is a BNE of the direct revelation mechanism $\Gamma = (\Theta_1, \ldots, \Theta_N, f)$.

That is, for all i, θ_i, and $\hat{\theta}_i \in \Theta_i$,

$$\mathbb{E}_{\theta_{-i}}\left[u_i(f(\theta_i, \theta_{-i}), \theta_i)|\theta_i\right] \geq \mathbb{E}_{\theta_{-i}}\left[u_i(f(\hat{\theta}_i, \theta_{-i}), \theta_i)|\theta_i\right].$$

Every player is asked for their type; reporting truthfully is a BNE.

Revelation principle

Theorem (Revelation principle for Bayes-Nash equilibrium)

If there exists a mechanism $\Gamma = (A_1, \ldots, A_N, g)$ that implements f in BNE, then f is truthfully implementable in BNE.

The proof is pretty much the same as for DSIC.