Mechanism Design

Lecture 13

Lecture Overview

Now that we've seen that voting doesn't work...

- ...let's talk about due dates:
  - Next Tuesday (March 11): assignment 2 due
  - Following Tuesday (March 18): midterm
  - Outline due: ...let's decide
  - Will the project allow work in pairs?
  - Assignment 3 out: probably April 1
  - Assignment 3 due: April 10 (last class)
  - Take-home exam: sometime between April 15 and 29 (48 hours)
  - Project due: ...let's decide
  - Latest possible date for all peer reviews of others' projects to be submitted: April 29

Lecture Overview

Notation

- N is the set of agents
- O is a finite set of outcomes with $|O| \geq 3$
- L is the set of all possible strict preference orderings over O.
  - for ease of exposition we switch to strict orderings
  - we will end up showing that desirable SWFs cannot be found *even if* preferences are restricted to strict orderings
- $[\succ]$ is an element of the set L^n (a preference ordering for every agent; the input to our social welfare function)
- $\succ_W$ is the preference ordering selected by the social welfare function W.
  - When the input to W is ambiguous we write it in the subscript; thus, the social order selected by W given the input $[\succ']$ is denoted as $\succ_{W([\succ'])}$.

Pareto Efficiency

Definition (Pareto Efficiency (PE))

W is Pareto efficient if for any $o_1, o_2 \in O$, $\forall i\, o_1 \succ_i o_2$ implies that $o_1 \succ_W o_2$.

- when all agents agree on the ordering of two outcomes, the social welfare function must select that ordering.

Independence of Irrelevant Alternatives

Definition (Independence of Irrelevant Alternatives (IIA))

W is independent of irrelevant alternatives if, for any $o_1, o_2 \in O$ and any two preference profiles $[\succ'], [\succ''] \in L^n$, $\forall i\, (o_1 \succ'_i o_2$ if and only if $o_1 \succ''_i o_2)$ implies that $(o_1 \succ_{W([\succ'])} o_2$ if and only if $o_1 \succ_{W([\succ''])} o_2)$.

- the selected ordering between two outcomes should depend only on the relative orderings they are given by the agents.

Nondictatorship

Definition (Non-dictatorship)

W does not have a dictator if $\neg\exists i\, \forall o_1, o_2 (o_1 \succ_i o_2 \Rightarrow o_1 \succ_W o_2)$.

- there does not exist a single agent whose preferences always determine the social ordering.
- We say that W is dictatorial if it fails to satisfy this property.

Arrow's Theorem

Theorem (Arrow, 1951)

Any social welfare function W that is Pareto efficient and independent of irrelevant alternatives is dictatorial.

We will assume that W is both PE and IIA, and show that W must be dictatorial. Our assumption that $|O| \geq 3$ is necessary for this proof. The argument proceeds in four steps.

Lecture Overview

Social Choice Functions

- Maybe Arrow's theorem held because we required a whole preference ordering.
- Idea: social choice functions might be easier to find
- We'll need to redefine our criteria for the social choice function setting; PE and IIA discussed the ordering

Weak Pareto Efficiency

Definition (Weak Pareto Efficiency)

A social choice function C is weakly Pareto efficient if, for any preference profile $[\succ] \in L^n$, if there exist a pair of outcomes o_1 and o_2 such that $\forall i \in N$, $o_1 \succ_i o_2$, then $C([\succ]) \neq o_2$.

- A dominated outcome can't be chosen.

Monotonicity

Definition (Monotonicity)

C is monotonic if, for any $o \in O$ and any preference profile $[\succ] \in L^n$ with $C([\succ]) = o$, then for any other preference profile $[\succ']$ with the property that $\forall i \in N, \forall o' \in O$, $o \succ'_i o'$ if $o \succ_i o'$, it must be that $C([\succ']) = o$.

- an outcome o must remain the winner whenever the support for it is increased relative to a preference profile under which o was already winning

Non-dictatorship

Definition (Non-dictatorship)

C is non-dictatorial if there does not exist an agent j such that C always selects the top choice in j's preference ordering.

The bad news

Theorem (Muller-Satterthwaite, 1977)

Any social choice function that is weakly Pareto efficient and monotonic is dictatorial.

- Perhaps contrary to intuition, social choice functions are no simpler than social welfare functions after all.
- The proof repeatedly "probes" a social choice function to determine the relative social ordering between given pairs of outcomes.
- Because the function must be defined for all inputs, we can use this technique to construct a full social welfare ordering.

Example: Plurality

Plurality satisfies weak PE and ND, so it must not be monotonic.

Consider the following preferences:

$$\begin{aligned} &\text{3 agents:} \quad a \succ b \succ c \\ &\text{2 agents:} \quad b \succ c \succ a \\ &\text{2 agents:} \quad c \succ b \succ a \end{aligned}$$

Plurality chooses a.

Example: Plurality

Plurality satisfies weak PE and ND, so it must not be monotonic.

Consider the following preferences:

3 agents: $a \succ b \succ c$
2 agents: $b \succ c \succ a$
2 agents: $c \succ b \succ a$

Plurality chooses a.

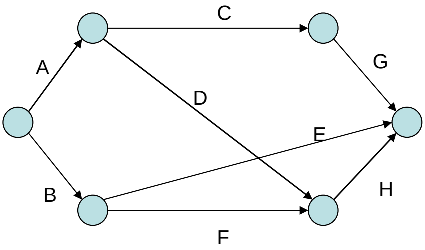
Increase support for a by moving c to the bottom:

3 agents: $a \succ b \succ c$
2 agents: $b \succ c \succ a$
2 agents: $b \succ a \succ c$

Now plurality chooses b.

Lecture Overview

Selfish Routing



- 8 people play as agents $A - H$; the others act as mediators.
- Agents' utility functions: $u_i =$ payment - cost if your edge is chosen; 0 otherwise.
- Mediators: find me a path from source to sink, at the lowest cost you can.
- Agents: agree to be paid whatever you like; claim whatever you like; however, you can't show your paper to anyone.

Lecture Overview

Bayesian Game Setting

- Extend the social choice setting to a new setting where agents can't be relied upon to disclose their preferences honestly.
- Start with a set of agents in a Bayesian game setting (but no actions).

Definition (Bayesian game setting)

A Bayesian game setting is a tuple (N, O, Θ, p, u), where

- N is a finite set of n agents;
- O is a set of outcomes;
- $\Theta = \Theta_1 \times \cdots \times \Theta_n$ is a set of possible joint type vectors;
- p is a (common prior) probability distribution on Θ; and
- $u = (u_1, \ldots, u_n)$, where $u_i : O \times \Theta \mapsto \mathbb{R}$ is the utility function for each player i.

Mechanism Design

Definition (Mechanism)

A mechanism (for a Bayesian game setting (N, O, Θ, p, u)) is a pair (A, M), where

- $A = A_1 \times \cdots \times A_n$, where A_i is the set of actions available to agent $i \in N$; and
- $M : A \mapsto \Pi(O)$ maps each action profile to a distribution over outcomes.

Thus, the designer gets to specify

- the action sets for the agents (though they may be constrained by the environment)
- the mapping to outcomes, over which agents have utility
- can't change outcomes; agents' preferences or type spaces

What we're up to

- The problem is to pick a mechanism that will cause rational agents to behave in a desired way, specifically maximizing the mechanism designer's own "utility" or objective function
  - each agent holds private information, in the Bayesian game sense
  - often, we're interested in settings where agents' action space is identical to their type space, and an action can be interpreted as a declaration of the agent's type
- Various equivalent ways of looking at this setting
  - perform an optimization problem, given that the values of (some of) the inputs are unknown
  - choose the Bayesian game out of a set of possible Bayesian games that maximizes some performance measure
  - design a game that *implements* a particular social choice function in equilibrium, given that the designer no longer knows agents' preferences and the agents might lie

Implementation in Dominant Strategies

Definition (Implementation in dominant strategies)

Given a Bayesian game setting (N, O, Θ, p, u), a mechanism (A, M) is an implementation in dominant strategies of a social choice function C (over N and O) if for any vector of utility functions u, the game has an equilibrium in dominant strategies, and in any such equilibrium a^* we have $M(a^*) = C(u)$.

Implementation in Bayes-Nash equilibrium

Definition (Bayes–Nash implementation)

Given a Bayesian game setting (N, O, Θ, p, u), a mechanism (A, M) is an implementation in Bayes–Nash equilibrium of a social choice function C (over N and O) if there exists a Bayes–Nash equilibrium of the game of incomplete information (N, A, Θ, p, u) such that for every $\theta \in \Theta$ and every action profile $a \in A$ that can arise given type profile θ in this equilibrium, we have that $M(a) = C(u(\cdot, \theta))$.

Bayes-Nash Implementation Comments

Bayes-Nash Equilibrium Problems:

- there could be more than one equilibrium
  - which one should I expect agents to play?
- agents could miscoordinate and play none of the equilibria
- asymmetric equilibria are implausible

Refinements:

- Symmetric Bayes-Nash implementation
- *Ex-post* implementation

Implementation Comments

We can require that the desired outcome arises

- in the only equilibrium
- in every equilibrium
- in at least one equilibrium

Forms of implementation:

- Direct Implementation: agents each simultaneously send a single message to the center
- Indirect Implementation: agents may send a sequence of messages; in between, information may be (partially) revealed about the messages that were sent previously like extensive form