Algorithmic Game Theory
COMP6207

Lecture 8: Dominant Strategy Implementation

Bahar Rastegari
b.rastegari@soton.ac.uk
Electronics and Computer Science
University of Southampton

Learning Outcomes

By the end of this session, you should be able to

- **Explain** what social choice functions can be implemented in dominant strategies.
- **Provide** characterisation of implementable social choice functions in both unrestricted quasilinear setting and single dimensional setting.

Recap

Revelation Principle

In assuming DS truthfulness, we make two separate assumptions:

1. every participant has a dominant strategy, and
2. this dominant strategy is direct revelation, where the participant truthfully reports all of its private information to the mechanism.

Revelation Principle

In assuming DS truthfulness, we make two separate assumptions:

1. every participant has a dominant strategy, and
2. this dominant strategy is direct revelation, where the participant truthfully reports all of its private information to the mechanism.

The revelation principle tells us that given (1), there is no need to relax requirement (2), it comes for free.

Any social choice function that can be implemented by any mechanism can be implemented by a direct, truthful mechanism.

Revelation Principle

In assuming DS truthfulness, we make two separate assumptions:

1. every participant has a dominant strategy, and
2. this dominant strategy is direct revelation, where the participant truthfully reports all of its private information to the mechanism.

The revelation principle tells us that given (1), there is no need to relax requirement (2), it comes for free.

Any social choice function that can be implemented by any mechanism can be implemented by a direct, truthful mechanism.

Theorem (Revelation Principle)

For every mechanism in which every participant has a dominant strategy (no matter what her private information), there is an equivalent direct dominant-strategy truthful mechanism.

Are indirect mechanisms ever useful?

- A direct truthful mechanism forces the agents to reveal their types completely. There might be settings where agents are not willing to compromise their privacy to this degree.
- Full revelation can sometimes place an unreasonable burden on the communication channel.
- Agents' equilibrium strategies might be difficult to compute; in this case the additional burden absorbed by the mechanism might be considerable.

What social choice functions can we implement in dominant strategies?

Let $X_i(\hat{v}_{-i}) \subseteq X$ denote the set of choices that can be selected by the choice rule χ given the declaration $\hat{v}_{-i}$ by the agents other than i.

Theorem

A mechanism is dominant-strategy truthful if and only if it satisfies the following conditions for every agent i and every $\hat{v}_{-i}$:

1. *The payment function $p_i(\hat{v})$ can be written as $p_i(\hat{v}_{-i}, \chi(\hat{v}))$.*
2. $\forall v_i$, $\chi(v_i, \hat{v}_{-i}) \in \operatorname{argmax}_{x \in X_i(\hat{v}_{-i})}(v_i(x) - p_i(\hat{v}_{-i}, x))$.

- An agent's payment can only depend on other agents' declarations and the selected choice
- The mechanism optimises for each agent: taking the other agents' declarations and the payment function into account, from every player's point of view the mechanism selects the most preferable choice.

What social choice functions can we implement in dominant strategies?

Let $X_i(\hat{v}_{-i}) \subseteq X$ denote the set of choices that can be selected by the choice rule χ given the declaration $\hat{v}_{-i}$ by the agents other than i.

Theorem

A mechanism is dominant-strategy truthful if and only if it satisfies the following conditions for every agent i and every $\hat{v}_{-i}$:

1. *The payment function $p_i(\hat{v})$ can be written as $p_i(\hat{v}_{-i}, \chi(\hat{v}))$.*
2. $\forall v_i$, $\chi(v_i, \hat{v}_{-i}) \in \operatorname{argmax}_{x \in X_i(\hat{v}_{-i})}(v_i(x) - p_i(\hat{v}_{-i}, x))$.

- An agent's payment can only depend on other agents' declarations and the selected choice
- The mechanism optimises for each agent: taking the other agents' declarations and the payment function into account, from every player's point of view the mechanism selects the most preferable choice.

We can replace $\hat{v}_{-i}$ with v_{-i} in above. Do you see why?

Implementable choice rules

What choice rules can we implement?

- The previous characterisation involves both the choice rule (i.e. social choice function) and the payment functions.
- What can we say about the "choice rules" ("social choice functions") that can be implemented?

What choice rules can we implement?

- The previous characterisation involves both the choice rule (i.e. social choice function) and the payment functions.
- What can we say about the "choice rules" ("social choice functions") that can be implemented?
- In what follows I am going to use "choice rule" and "social choice function" interchangeably.

Weak Monotonicity

Definition (Weak Monotonicity)

A social choice function χ satisfies weak monotonicity (WMON) if for all agents i and all possible valuation profiles of the other agents v_{-i} we have that $\chi(v_i, v_{-i}) = x \neq y = \chi(v_i', v_{-i})$ implies that $v_i'(y) - v_i(y) \geq v_i'(x) - v_i(x)$.

- WMON means that if the social choice changes when a single agent changes her valuation, then it must be that the increase in her value for the new choice (i.e. $v_i'(y) - v_i(y)$) is at least as large as the increase in her value for the old choice (i.e. $v_i'(x) - v_i(x)$).

WMON is Necessary

Theorem (WMON is Necessary)

If a mechanism (χ, p) is DS truthful, then χ satisfies WMON.

WMON is Necessary

Theorem (WMON is Necessary)

If a mechanism (χ, p) is DS truthful, then χ satisfies WMON.

Theorem (When WMON is Sufficient)

If all domains of preferences V_i are convex sets (as subsets of Euclidean space) then for every social choice function χ that satisfies WMON there exists payment functions $p_1, \ldots, p_n$ such that (χ, p) is DS truthful.

Two special cases

Question: what social choice functions are WMON?

Two special cases

Question: what social choice functions are WMON?

We don't know the answer in general, but we do know it for some extreme cases. Here we will consider two:

1. Unrestricted quasilinear settings where $V_i = \mathbb{R}^X$
2. Single dimensional settings

The case for unrestricted quasilinear settings

Affine Maximiser or Weighted VCG

Definition (Affine maximiser)

A social choice function χ is an affine maximiser if for some subrange $X' \subseteq X$, for some agent weights $w_i \in \mathbb{R}^+$, and for some outcome weights $\gamma_x \in \mathbb{R}$, $\forall x \in X'$, χ has the form

$$\underset{x \in X'}{\operatorname{argmax}}(\gamma_x + \sum_i w_i v_i(x)).$$

Affine Maximiser or Weighted VCG

Definition (Affine maximiser)

A social choice function χ is an affine maximiser if for some subrange $X' \subseteq X$, for some agent weights $w_i \in \mathbb{R}^+$, and for some outcome weights $\gamma_x \in \mathbb{R}$, $\forall x \in X'$, χ has the form

$$\operatorname*{argmax}_{x \in X'}(\gamma_x + \sum_i w_i v_i(x)).$$

Exercise: Any affine maximiser social choice function χ can be implemented in dominant strategies; i.e. there exist payment functions $p_1, \ldots, p_n$ where (χ, p) is DS truthful.

Affine Maximiser

Definition (Affine maximiser)

A social choice function χ is an affine maximiser if for some subrange $X' \subseteq X$, for some agent weights $w_i \in \mathbb{R}^+$, and for some outcome weights $\gamma_x \in \mathbb{R}$, $\forall x \in X'$, χ has the form

$$\operatorname*{argmax}_{x \in X'} (\gamma_x + \sum_i w_i v_i(x)).$$

Theorem (Roberts)

If there are at least three choices that a social choice function will choose given some input, and if agents have general quasilinear preferences, then the set of (deterministic) social choice functions implementable in dominant strategies is precisely the set of affine maximisers.

Understanding Roberts

- In the case of general quasilinear preferences (i.e., when each agent can have any valuation for each choice $x \in X$) and where the choice function selects from more than two alternatives, affine maximisers are the only DS-implementable social choice functions.

Understanding Roberts

- In the case of general quasilinear preferences (i.e., when each agent can have any valuation for each choice $x \in X$) and where the choice function selects from more than two alternatives, affine maximisers are the only DS-implementable social choice functions.
- Efficiency is an affine-maximising social choice function for which $\forall x \in X, \gamma_x = 0$ and $\forall i \in N, w_i = 1$.
  - Affine maximising mechanisms are weighted Groves mechanisms.
    - They transform both the choices and the agents' valuations by applying linear weights, then effectively run a Groves mechanism in the transformed space.
  - Thus, we cannot stray very far from Groves mechanisms even if we give up on efficiency.

Understanding Roberts

- In the case of general quasilinear preferences (i.e., when each agent can have any valuation for each choice $x \in X$) and where the choice function selects from more than two alternatives, affine maximisers are the only DS-implementable social choice functions.
- Efficiency is an affine-maximising social choice function for which $\forall x \in X, \gamma_x = 0$ and $\forall i \in N, w_i = 1$.
  - Affine maximising mechanisms are weighted Groves mechanisms.
    - They transform both the choices and the agents' valuations by applying linear weights, then effectively run a Groves mechanism in the transformed space.
  - Thus, we cannot stray very far from Groves mechanisms even if we give up on efficiency.
- It is possible to implement a richer set of functions when agents' preferences are restricted further.

The case for

single dimensional settings

Single-parameter domains

- The opposite case to full dimensionality of unrestricted quasilinear
- There is a single real parameter that directly determines the whole valuation vector v_i.
- There are several possible levels of generality in which to formalize this, we will consider an intermediate level that is simple and yet sufficient for most applications.

Single-parameter domains

- The opposite case to full dimensionality of unrestricted quasilinear
- There is a single real parameter that directly determines the whole valuation vector v_i.
- There are several possible levels of generality in which to formalize this, we will consider an intermediate level that is simple and yet sufficient for most applications.
- In our setting, each bidder has a scalar value for "winning", with "losing" having value of 0.
- Each agent is associated with a subset of winning alternatives $W_i \subseteq X$ where all $x \in W_i$ are equivalent to each others (in terms of their value) to agent i, and all $x \notin W_i$ are valued at 0 by i.

Examples

- Single-item auction where (as we have assumed in this module) agents only care about whether they receive the item or not. So W_i is the single outcome where i wins.

Examples

- Single-item auction where (as we have assumed in this module) agents only care about whether they receive the item or not. So W_i is the single outcome where i wins.

- A combinatorial auction setting with so called "known single-minded" agents, where each agent is only interested in a specific bundle of items (that is known to the mechanism), with only her value for the bundle being her private information. So W_i are all the outcomes in which i receives her bundle of interest.

Examples

- Single-item auction where (as we have assumed in this module) agents only care about whether they receive the item or not. So W_i is the single outcome where i wins.
- A combinatorial auction setting with so called "known single-minded" agents, where each agent is only interested in a specific bundle of items (that is known to the mechanism), with only her value for the bundle being her private information. So W_i are all the outcomes in which i receives her bundle of interest.
- Selfish routing setting where the mechanism wants to buy a path.

Examples

- Single-item auction where (as we have assumed in this module) agents only care about whether they receive the item or not. So W_i is the single outcome where i wins.
- A combinatorial auction setting with so called "known single-minded" agents, where each agent is only interested in a specific bundle of items (that is known to the mechanism), with only her value for the bundle being her private information. So W_i are all the outcomes in which i receives her bundle of interest.
- Selfish routing setting where the mechanism wants to buy a path. W_i is the set of all paths that contain edge i.

Single-parameter domains

Setting:

- Each agent i is associated with a subset of winning alternatives $W_i \subseteq X$, and a range of values $[t^0, t^1]$, which are both publicly known.
- Agent i values all $x \in W_i$ at v_i (which is her private knowledge), and has value of 0 for all other choices.

Single-parameter domains

Setting:

- Each agent i is associated with a subset of winning alternatives $W_i \subseteq X$, and a range of values $[t^0, t^1]$, which are both publicly known.
- Agent i values all $x \in W_i$ at v_i (which is her private knowledge), and has value of 0 for all other choices.

Definition (Monotonicity in single-parameter domains)

A social choice function χ on a single parameter domain is called monotone in v_i if for every v_{-i} and every $v_i' \geq v_i$ we have that $\chi(v_i, v_{-i}) \in W_i$ implies that $\chi(v_i', v_{-i}) \in W_i$. That is, if valuation v_i makes i win, then so will every higher valuation $v_i' \geq v_i$.

Critical values

- For a monotone function χ, for every v_{-i} for which agent i can both win and lose, there is always a critical value $c_i(v_{-i})$ at and below which i loses and above which she wins.

Critical values

- For a monotone function χ, for every v_{-i} for which agent i can both win and lose, there is always a critical value $c_i(v_{-i})$ at and below which i loses and above which she wins.

- If i wins for every v_i then the critical value at v_{-i} is undefined.

Normalised mechanisms

- A mechanism on single parameter domain is called normalised if the payment for losing is always 0. That is, $p_i(v) = 0$ if $\chi(v_i, v_{-i}) \notin W_i$.

Normalised mechanisms

- A mechanism on single parameter domain is called normalised if the payment for losing is always 0. That is, $p_i(v) = 0$ if $\chi(v_i, v_{-i}) \notin W_i$.

- Every DS truthful mechanism can be easily turned into a normalised one. So it suffices to characterise normalised mechanisms.

Characterising implementable social choice functions

Theorem

A normalised mechanism (χ, p) on a single parameter domain is DS truthful if and only if the following conditions hold:

1. *χ is monotone in every v_i*
2. *Every winning bid pays the critical value.*
   - *If $\chi(v_i, v_{-i}) \in W_i$ then $p_i(v_i, v_{-i}) = c_i(v_{-i})$.*
   - *If $c_i(v_{-i})$ is undefined (implying that given v_{-i}, i wins for all v_i), then there exists some value r_i, $r_i \leq t^0$, such that $p_i(v_i, v_{-i}) = r_i$.*

Characterising implementable social choice functions

Theorem

A normalised mechanism (χ, p) on a single parameter domain is DS truthful if and only if the following conditions hold:

1. χ *is monotone in every* v_i
2. *Every winning bid pays the critical value.*
   - *If* $\chi(v_i, v_{-i}) \in W_i$ *then* $p_i(v_i, v_{-i}) = c_i(v_{-i})$.
   - *If* $c_i(v_{-i})$ *is undefined (implying that given* v_{-i}, i *wins for all* v_i*), then there exists some value* r_i, $r_i \leq t^0$, *such that* $p_i(v_i, v_{-i}) = r_i$.

Understanding this theorem:

- Not restricted to affine-maximisers
- E.g. $\chi = \text{argmax}_x \sum_i v_i(x)^2$, or $\chi = \text{argmax}_x\, min_i v_i(x)$

Characterising implementable social choice functions

Theorem

A normalised mechanism (χ, p) on a single parameter domain is DS truthful if and only if the following conditions hold:

1. χ *is monotone in every* v_i
2. *Every winning bid pays the critical value.*
   - *If* $\chi(v_i, v_{-i}) \in W_i$ *then* $p_i(v_i, v_{-i}) = c_i(v_{-i})$.
   - *If* $c_i(v_{-i})$ *is undefined (implying that given* v_{-i}, i *wins for all* v_i*), then there exists some value* r_i, $r_i \leq t^0$, *such that* $p_i(v_i, v_{-i}) = r_i$.

Understanding this theorem:

- Not restricted to affine-maximisers
- E.g. $\chi = \text{argmax}_x \sum_i v_i(x)^2$, or $\chi = \text{argmax}_x \, min_i v_i(x)$
- In many cases this flexibility allows the design of tractable (i.e. computationally efficient) approximation mechanisms for problems whose exact optimisation is computationally intractable.

Computational applications of mechanism design

- Task scheduling: allocate tasks among agents to minimise makespan

Computational applications of mechanism design

- Task scheduling: allocate tasks among agents to minimise makespan
- Bandwidth allocation in computer networks: allocate real-valued capacity of a single network link among users with different demand curves.

Computational applications of mechanism design

- Task scheduling: allocate tasks among agents to minimise makespan
- Bandwidth allocation in computer networks: allocate real-valued capacity of a single network link among users with different demand curves.
- Multicast cost sharing: share the cost of a multicast transmission among the users who receive it

Computational applications of mechanism design

- Task scheduling: allocate tasks among agents to minimise makespan
- Bandwidth allocation in computer networks: allocate real-valued capacity of a single network link among users with different demand curves.
- Multicast cost sharing: share the cost of a multicast transmission among the users who receive it
- Two-sided matching: pair up members of two groups according to their preferences, without imposing any payments, e.g. students and supervisors, hospitals and residents, kidney donors and recipients.

Books

- Twenty Lectures on Algorithmic Game Theory, by Tim Roughgarden
- Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations by Yoav Shoham and Kevin Leyton-Brown
  - From now on we will refer to this book as **MAS**
- Algorithmic Game Theory, edited by Noam Nisan, Tim Roughgarden, Eva Tardos,Vijay V. Vazirani
  - From now on we will refer to this book as **AGT**

Further reading/watching

- Read MAS chapter 10.5.1
- Read AGT Chapters 9.5.2, 9.5.3, 9.5.4