

भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay

CS 6001: Game Theory and Algorithmic Mechanism Design

Week 11 – Affine Maximizers, Myerson's Lemma, and Bayesian Mechanisms

Swaprava Nath

October, 2026

ज्ञानम् परमम् ध्येयम्

Knowledge is the supreme goal

Week 11 Roadmap

Lecture	Chunks (20–25 min each)	Quizzes
Lecture 21	1. Affine Maximizers 2. INA, and AMs are DSIC 3. Roberts' Theorem and Single Object Allocation 4. A Toolkit from Convex Analysis	Quiz 1 after Chunk 2 Quiz 2 after Chunk 4
Lecture 22	1. Myerson's Lemma: Forward Direction 2. Reverse Direction and Examples 3. Individual Rationality and Non-Vickrey Auctions 4. Bayesian Mechanisms: BIC, NDE, and IIR	Quiz 1 after Chunk 2 Quiz 2 after Chunk 4

Contents



- Lecture 21: Affine Maximizers and Single Object Allocation
- Lecture 22: Myerson's Lemma and Optimal Mechanism Design

Chunk 1

Affine Maximizers

(≈ 20–25 minutes)

Where We Stand

- VCG is truthful. It is also elegant.

Where We Stand

- VCG is truthful. It is also elegant.
- But it has three practical pain points.

Where We Stand

- VCG is truthful. It is also elegant.
- But it has three practical pain points.
  - It is **not budget balanced**.

Where We Stand

- VCG is truthful. It is also elegant.
- But it has three practical pain points.
  - It is **not budget balanced**.
  - It treats all agents **symmetrically**. Sometimes we do not want that.

Where We Stand

- VCG is truthful. It is also elegant.
- But it has three practical pain points.
  - It is **not budget balanced**.
  - It treats all agents **symmetrically**. Sometimes we do not want that.
  - It always picks the **efficient** outcome. Sometimes the designer wants something else.

Where We Stand

- VCG is truthful. It is also elegant.
- But it has three practical pain points.
  - It is **not budget balanced**.
  - It treats all agents **symmetrically**. Sometimes we do not want that.
  - It always picks the **efficient** outcome. Sometimes the designer wants something else.
- Today: a bigger family. Give **weights** to agents. Give a **bonus** to outcomes.

Generalization of VCG mechanism

Question

Can we incorporate a larger class of DSIC mechanisms in the quasi-linear domain?

Generalization of VCG mechanism

Question

Can we incorporate a larger class of DSIC mechanisms in the quasi-linear domain?

Definition (Affine Maximizer (AM) Allocation Rule)

$$w_i \theta_i(a)$$

where, $w_i \geqslant 0 \quad \forall i \in N$, (not all zero) – **potentially different weights for players**

Generalization of VCG mechanism

Question

Can we incorporate a larger class of DSIC mechanisms in the quasi-linear domain?

Definition (Affine Maximizer (AM) Allocation Rule)

$$\sum_{i \in N} w_i \theta_i(a)$$

where, $w_i \geqslant 0 \quad \forall i \in N$, (not all zero) – **potentially different weights for players**

Generalization of VCG mechanism

Question

Can we incorporate a larger class of DSIC mechanisms in the quasi-linear domain?

Definition (Affine Maximizer (AM) Allocation Rule)

$$\sum_{i \in N} w_i \theta_i(a) + \kappa(a)$$

where, $w_i \geqslant 0 \quad \forall i \in N$, (not all zero) – **potentially different weights for players**
$\kappa : A \to \mathbb{R}$ is any arbitrary function – **translation**

Generalization of VCG mechanism

Question

Can we incorporate a larger class of DSIC mechanisms in the quasi-linear domain?

Definition (Affine Maximizer (AM) Allocation Rule)

$$\arg\max_{a \in A} \left(\sum_{i \in N} w_i \theta_i(a) + \kappa(a) \right)$$

where, $w_i \geqslant 0 \quad \forall i \in N$, (not all zero) – **potentially different weights for players**
$\kappa : A \to \mathbb{R}$ is any arbitrary function – **translation**

Generalization of VCG mechanism

Question

Can we incorporate a larger class of DSIC mechanisms in the quasi-linear domain?

Definition (Affine Maximizer (AM) Allocation Rule)

$$f^{AM}(\theta) \in \arg\max_{a \in A} \left(\sum_{i \in N} w_i \theta_i(a) + \kappa(a) \right)$$

where, $w_i \geqslant 0 \quad \forall i \in N$, (not all zero) – **potentially different weights for players**
$\kappa : A \to \mathbb{R}$ is any arbitrary function – **translation**

Generalization of VCG mechanism

Question

Can we incorporate a larger class of DSIC mechanisms in the quasi-linear domain?

Definition (Affine Maximizer (AM) Allocation Rule)

$$f^{AM}(\theta) \in \arg\max_{a \in A} \left(\sum_{i \in N} w_i \theta_i(a) + \kappa(a) \right)$$

where, $w_i \geqslant 0 \ \ \forall i \in N$, (not all zero) – **potentially different weights for players**
$\kappa : A \to \mathbb{R}$ is any arbitrary function – **translation**

Special cases

- $\kappa \equiv 0$ and $w_i = 1 \ \ \forall i \in N$ – **efficient**
- $\kappa \equiv 0$ and $w_d = 1, w_i = 0 \ \ \forall i \neq d$ – **dictatorial**
- w_i's are different $\implies$ not ANON
- κ is a non-constant function $\implies$ different importance is given to different allocations

The Surge Policy Vote

A ride-hailing app picks **one** city-wide pricing policy. Two voices matter.



	Surge	Flat	Hybrid
θ_1 (drivers)	6	1	3
θ_2 (riders)	1	4	2

The Surge Policy Vote

A ride-hailing app picks **one** city-wide pricing policy. Two voices matter.



	Surge	Flat	Hybrid
θ_1 (drivers)	6	1	3
θ_2 (riders)	1	4	2

- Equal weights, $\kappa \equiv 0$: scores $7, 5, 5$. **Surge** wins. This is efficient (VCG).

The Surge Policy Vote

A ride-hailing app picks **one** city-wide pricing policy. Two voices matter.



	Surge	Flat	Hybrid
θ_1 (drivers)	6	1	3
θ_2 (riders)	1	4	2

- Equal weights, $\kappa \equiv 0$: scores $7, 5, 5$. **Surge** wins. This is efficient (VCG).
- Riders get weight $w_2 = 2$: scores $8, 9, 7$. **Flat** wins. Not anonymous anymore.

The Surge Policy Vote

A ride-hailing app picks **one** city-wide pricing policy. Two voices matter.

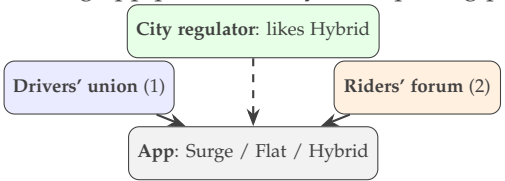


	Surge	Flat	Hybrid
θ_1 (drivers)	6	1	3
θ_2 (riders)	1	4	2

- Equal weights, $\kappa \equiv 0$: scores $7, 5, 5$. **Surge** wins. This is efficient (VCG).
- Riders get weight $w_2 = 2$: scores $8, 9, 7$. **Flat** wins. Not anonymous anymore.
- *Swap the names of the two agents, and the outcome changes.*

The Surge Policy Vote

A ride-hailing app picks **one** city-wide pricing policy. Two voices matter.



	Surge	Flat	Hybrid
θ_1 (drivers)	6	1	3
θ_2 (riders)	1	4	2
κ (regulator)	0	0	3

- Equal weights, $\kappa \equiv 0$: scores $7, 5, 5$. **Surge** wins. This is efficient (VCG).
- Riders get weight $w_2 = 2$: scores $8, 9, 7$. **Flat** wins. Not anonymous anymore.
- *Swap the names of the two agents, and the outcome changes.*
- Add the regulator's bonus $\kappa(\text{Hybrid}) = 3$: scores $8, 9, 10$. **Hybrid** wins.

Observations on Affine Maximizers

- Affine Maximizer is a super class of VCG (efficient) allocations, hence, it can satisfy more properties

Observations on Affine Maximizers

- Affine Maximizer is a super class of VCG (efficient) allocations, hence, it can satisfy more properties
- We can ask a characterization question (like GS Theorem) in the quasi-linear setting with public goods.

Observations on Affine Maximizers

- Affine Maximizer is a super class of VCG (efficient) allocations, hence, it can satisfy more properties
- We can ask a characterization question (like GS Theorem) in the quasi-linear setting with public goods.
- **Independence of Non-influential Agents**

Observations on Affine Maximizers

- Affine Maximizer is a super class of VCG (efficient) allocations, hence, it can satisfy more properties
- We can ask a characterization question (like GS Theorem) in the quasi-linear setting with public goods.
- **Independence of Non-influential Agents**

Definition

An AM rule f^{AM} with weights $w_i \ \forall i \in N$ and the function κ satisfies independence of non-influential agents (INA) if for all $i \in N$ with $w_i = 0$ we have

$$f^{AM}(\theta_i, \theta_{-i}) = f^{AM}(\theta_i', \theta_{-i}), \quad \forall\ \theta_i, \theta_i', \theta_{-i}$$

Observations on Affine Maximizers

- Affine Maximizer is a super class of VCG (efficient) allocations, hence, it can satisfy more properties
- We can ask a characterization question (like GS Theorem) in the quasi-linear setting with public goods.
- **Independence of Non-influential Agents**

Definition

An AM rule f^{AM} with weights $w_i \ \forall i \in N$ and the function κ satisfies independence of non-influential agents (INA) if for all $i \in N$ with $w_i = 0$ we have

$$f^{AM}(\theta_i, \theta_{-i}) = f^{AM}(\theta_i', \theta_{-i}), \quad \forall\ \theta_i, \theta_i', \theta_{-i}$$

- **Remark:** This is a tie-breaking requirement – the zero weight agent does not influence the allocation decision, hence it should not break any tie either



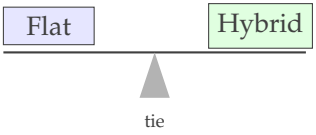
Chunk 2

INA, and Affine Maximizers are DSIC

(≈ 20–25 minutes)

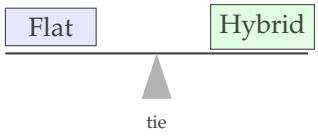
The Silent Auditor

- The app adds agent 3: a city **auditor bot**.



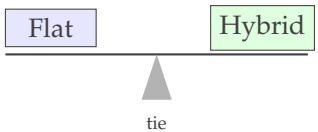
The Silent Auditor

- The app adds agent 3: a city **auditor bot**.
- Its weight is $w_3 = 0$. It is only there to watch.



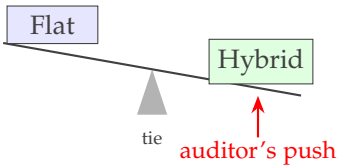
The Silent Auditor

- The app adds agent 3: a city **auditor bot**.
- Its weight is $w_3 = 0$. It is only there to watch.
- One day, Flat and Hybrid **tie** on the weighted sum.



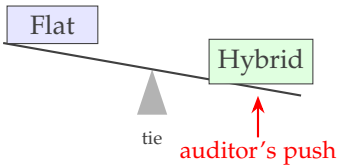
The Silent Auditor

- The app adds agent 3: a city **auditor bot**.
- Its weight is $w_3 = 0$. It is only there to watch.
- One day, Flat and Hybrid **tie** on the weighted sum.
- The app says: "let the auditor's reported less-preferred alternative break the tie."



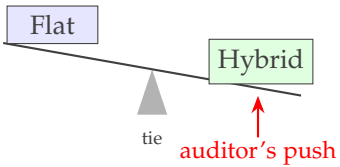
The Silent Auditor

- The app adds agent 3: a city **auditor bot**.
- Its weight is $w_3 = 0$. It is only there to watch.
- One day, Flat and Hybrid **tie** on the weighted sum.
- The app says: "let the auditor's reported less-preferred alternative break the tie."
- Now the auditor has a lever. It can **lie** to pull the outcome its way.



The Silent Auditor

- The app adds agent 3: a city **auditor bot**.
- Its weight is $w_3 = 0$. It is only there to watch.
- One day, Flat and Hybrid **tie** on the weighted sum.
- The app says: "let the auditor's reported less-preferred alternative break the tie."
- Now the auditor has a lever. It can **lie** to pull the outcome its way.
- A zero-weight agent should never hold a lever. That is **INA**.



(Almost) All Affine Maximizers are DSIC

Example

If INA was not satisfied, then AM can be manipulated, e.g., suppose there is a tie when $w_i = 0$ for some valuation profile, but the allocation is the less preferred one for agent i

(Almost) All Affine Maximizers are DSIC

Example

If INA was not satisfied, then AM can be manipulated, e.g., suppose there is a tie when $w_i = 0$ for some valuation profile, but the allocation is the less preferred one for agent i

Theorem

An AM rule satisfying INA is implementable in dominant strategies.

(Almost) All Affine Maximizers are DSIC

Example

If INA was not satisfied, then AM can be manipulated, e.g., suppose there is a tie when $w_i = 0$ for some valuation profile, but the allocation is the less preferred one for agent i

Theorem

An AM rule satisfying INA is implementable in dominant strategies.

Proof.

We need to construct a payment function p^{AM} to make (f^{AM}, p^{AM}) DSIC.

(Almost) All Affine Maximizers are DSIC

Example

If INA was not satisfied, then AM can be manipulated, e.g., suppose there is a tie when $w_i = 0$ for some valuation profile, but the allocation is the less preferred one for agent i

Theorem

An AM rule satisfying INA is implementable in dominant strategies.

Proof.

We need to construct a payment function p^{AM} to make (f^{AM}, p^{AM}) DSIC.
Consider

$$p_i^{AM}(\theta_i, \theta_{-i}) = \begin{cases} \frac{1}{w_i}[h_i(\theta_{-i}) - \{\sum_{j \neq i} w_j \theta_j(f^{AM}(\theta)) + \kappa(f^{AM}(\theta))\}] & \forall i : w_i > 0, \\ 0, & \forall i : w_i = 0. \end{cases}$$

□

Reading the Payment

$$p_i^{AM}(\theta_i, \theta_{-i}) = \begin{cases} \frac{1}{w_i}[h_i(\theta_{-i}) - \{\sum_{j \neq i} w_j \theta_j(f^{AM}(\theta)) + \kappa(f^{AM}(\theta))\}] & \forall i : w_i > 0, \\ 0, & \forall i : w_i = 0. \end{cases}$$

- $h_i(\theta_{-i})$: anything that does **not** depend on i's report.

Reading the Payment

$$p_i^{AM}(\theta_i, \theta_{-i}) = \begin{cases} \frac{1}{w_i}[h_i(\theta_{-i}) - \{\sum_{j \neq i} w_j \theta_j(f^{AM}(\theta)) + \kappa(f^{AM}(\theta))\}] & \forall i : w_i > 0, \\ 0, & \forall i : w_i = 0. \end{cases}$$

- $h_i(\theta_{-i})$: anything that does **not** depend on i's report.
- The bracket: how the **others** (and the regulator) fare at the chosen outcome.

Reading the Payment

$$p_i^{AM}(\theta_i, \theta_{-i}) = \begin{cases} \frac{1}{w_i}[h_i(\theta_{-i}) - \{\sum_{j \neq i} w_j \theta_j(f^{AM}(\theta)) + \kappa(f^{AM}(\theta))\}] & \forall i : w_i > 0, \\ 0, & \forall i : w_i = 0. \end{cases}$$

- $h_i(\theta_{-i})$: anything that does **not** depend on i's report.
- The bracket: how the **others** (and the regulator) fare at the chosen outcome.
- Divide by w_i: convert "weighted points" back to money.

Reading the Payment

$$p_i^{AM}(\theta_i, \theta_{-i}) = \begin{cases} \frac{1}{w_i}[h_i(\theta_{-i}) - \{\sum_{j \neq i} w_j \theta_j(f^{AM}(\theta)) + \kappa(f^{AM}(\theta))\}] & \forall i : w_i > 0, \\ 0, & \forall i : w_i = 0. \end{cases}$$

- $h_i(\theta_{-i})$: anything that does **not** depend on i's report.
- The bracket: how the **others** (and the regulator) fare at the chosen outcome.
- Divide by w_i: convert "weighted points" back to money.
- With $w_i = 1$ and $\kappa \equiv 0$, this is exactly the **Groves** payment.

Reading the Payment

$$p_i^{AM}(\theta_i, \theta_{-i}) = \begin{cases} \frac{1}{w_i}[h_i(\theta_{-i}) - \{\sum_{j \neq i} w_j \theta_j(f^{AM}(\theta)) + \kappa(f^{AM}(\theta))\}] & \forall i : w_i > 0, \\ 0, & \forall i : w_i = 0. \end{cases}$$

- $h_i(\theta_{-i})$: anything that does **not** depend on i's report.
- The bracket: how the **others** (and the regulator) fare at the chosen outcome.
- Divide by w_i: convert "weighted points" back to money.
- With $w_i = 1$ and $\kappa \equiv 0$, this is exactly the **Groves** payment.
- The trick is the same as Groves. Make i's payoff equal to the **whole AM objective**, scaled by $\frac{1}{w_i}$.

Reading the Payment

$$p_i^{AM}(\theta_i, \theta_{-i}) = \begin{cases} \frac{1}{w_i}[h_i(\theta_{-i}) - \{\sum_{j \neq i} w_j \theta_j(f^{AM}(\theta)) + \kappa(f^{AM}(\theta))\}] & \forall i : w_i > 0, \\ 0, & \forall i : w_i = 0. \end{cases}$$

- $h_i(\theta_{-i})$: anything that does **not** depend on i's report.
- The bracket: how the **others** (and the regulator) fare at the chosen outcome.
- Divide by w_i: convert "weighted points" back to money.
- With $w_i = 1$ and $\kappa \equiv 0$, this is exactly the **Groves** payment.
- The trick is the same as Groves. Make i's payoff equal to the **whole AM objective**, scaled by $\frac{1}{w_i}$.
- Then lying can only move the objective **down**.

(Almost) All Affine Maximizers are DSIC (contd.)



Proof.

Payoff of i if $w_i > 0$

(Almost) All Affine Maximizers are DSIC (contd.)

Proof.

Payoff of i if $w_i > 0$

$$
\begin{aligned}
&= \theta_i(f^{AM}(\theta_i, \theta_{-i})) - p_i^{AM}(\theta_i, \theta_{-i}) \\
&= \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i, \theta_{-i})) + \kappa(f^{AM}(\theta_i, \theta_{-i}))\} - h_i(\theta_{-i})] \\
&\geqslant \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i', \theta_{-i})) + \kappa(f^{AM}(\theta_i', \theta_{-i}))\} - h_i(\theta_{-i})] \\
&= \theta_i(f^{AM}(\theta_i', \theta_{-i})) - p_i^{AM}(\theta_i', \theta_{-i})
\end{aligned}
$$

(Almost) All Affine Maximizers are DSIC (contd.)

Proof.

Payoff of i if $w_i > 0$

$$
\begin{aligned}
&= \theta_i(f^{AM}(\theta_i, \theta_{-i})) - p_i^{AM}(\theta_i, \theta_{-i}) \\
&= \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i, \theta_{-i})) + \kappa(f^{AM}(\theta_i, \theta_{-i}))\} - h_i(\theta_{-i})] \\
&\geqslant \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i', \theta_{-i})) + \kappa(f^{AM}(\theta_i', \theta_{-i}))\} - h_i(\theta_{-i})] \\
&= \theta_i(f^{AM}(\theta_i', \theta_{-i})) - p_i^{AM}(\theta_i', \theta_{-i})
\end{aligned}
$$

For i where $w_i = 0$, payments are zero and

(Almost) All Affine Maximizers are DSIC (contd.)

Proof.

Payoff of i if $w_i > 0$

$$
\begin{aligned}
&= \theta_i(f^{AM}(\theta_i, \theta_{-i})) - p_i^{AM}(\theta_i, \theta_{-i}) \\
&= \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i, \theta_{-i})) + \kappa(f^{AM}(\theta_i, \theta_{-i}))\} - h_i(\theta_{-i})] \\
&\geqslant \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i', \theta_{-i})) + \kappa(f^{AM}(\theta_i', \theta_{-i}))\} - h_i(\theta_{-i})] \\
&= \theta_i(f^{AM}(\theta_i', \theta_{-i})) - p_i^{AM}(\theta_i', \theta_{-i})
\end{aligned}
$$

For i where $w_i = 0$, payments are zero and

$$f^{AM}(\theta_i, \theta_{-i}) = f^{AM}(\theta_i', \theta_{-i}),\ \forall \theta_i, \theta_i', \theta_{-i}$$

(Almost) All Affine Maximizers are DSIC (contd.)

Proof.

Payoff of i if $w_i > 0$

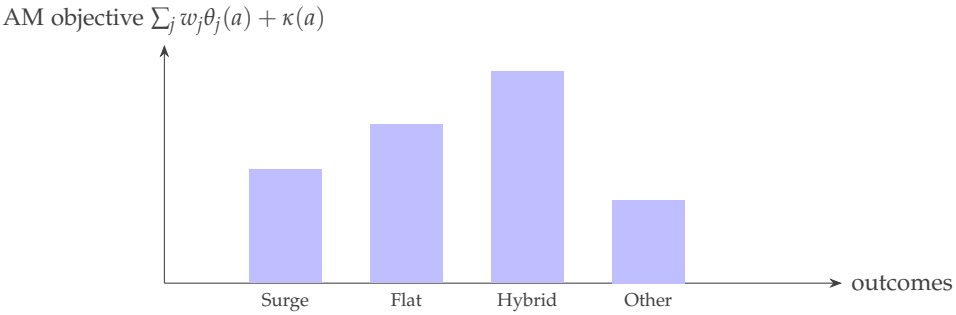
$$
\begin{aligned}
&= \theta_i(f^{AM}(\theta_i, \theta_{-i})) - p_i^{AM}(\theta_i, \theta_{-i}) \\
&= \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i, \theta_{-i})) + \kappa(f^{AM}(\theta_i, \theta_{-i}))\} - h_i(\theta_{-i})] \\
&\geqslant \frac{1}{w_i}[\{\sum_{j \in N} w_j \theta_j(f^{AM}(\theta_i', \theta_{-i})) + \kappa(f^{AM}(\theta_i', \theta_{-i}))\} - h_i(\theta_{-i})] \\
&= \theta_i(f^{AM}(\theta_i', \theta_{-i})) - p_i^{AM}(\theta_i', \theta_{-i})
\end{aligned}
$$

For i where $w_i = 0$, payments are zero and

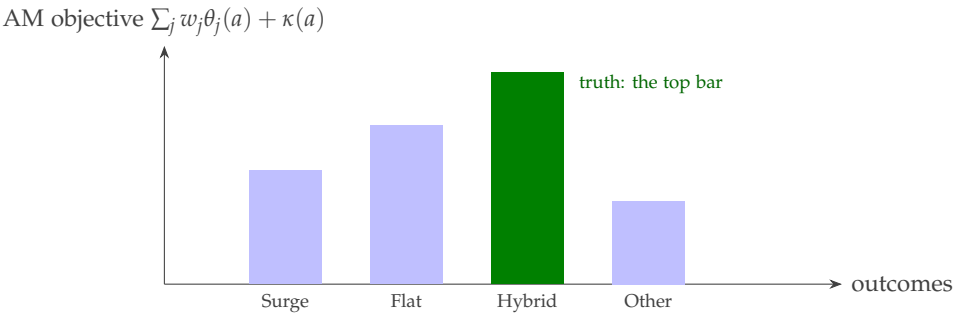
$$f^{AM}(\theta_i, \theta_{-i}) = f^{AM}(\theta_i', \theta_{-i}),\ \forall \theta_i, \theta_i', \theta_{-i}$$

Hence, payoff are identical for all types. □

Where the Inequality Comes From

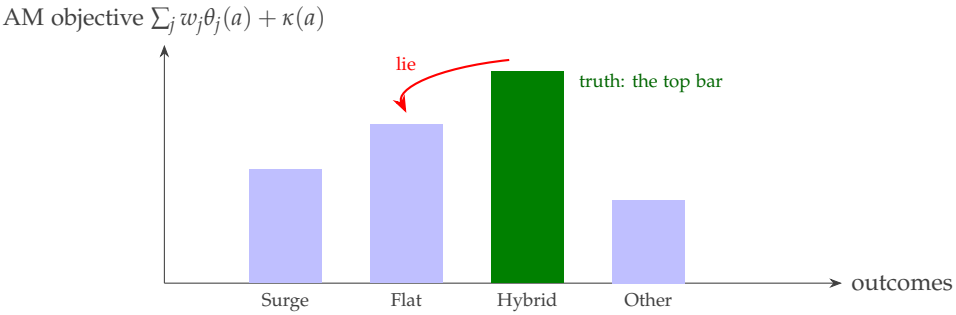


Where the Inequality Comes From



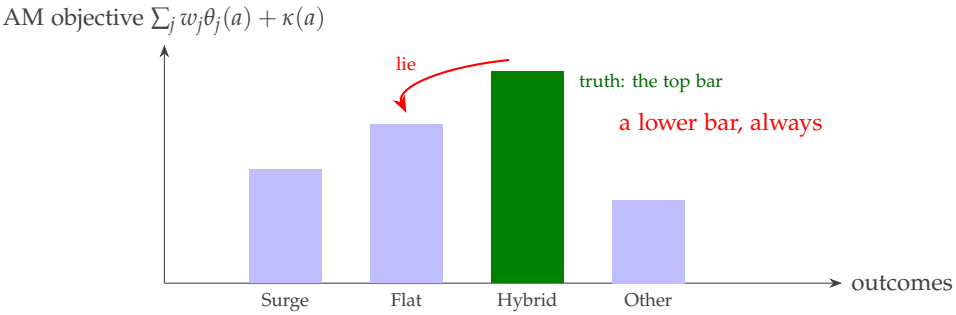
- Truthful reports make the AM pick the **tallest** bar.

Where the Inequality Comes From



- Truthful reports make the AM pick the **tallest** bar.
- A lie can only make the mechanism pick some other bar.

Where the Inequality Comes From



- Truthful reports make the AM pick the **tallest** bar.
- A lie can only make the mechanism pick some other bar.
- Agent i's payoff is $\frac{1}{w_i} \times$(bar height) $-$ constant. So lying never helps.

In-Class Quiz 1

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.



Chunk 3

Roberts' Theorem and Single Object Allocation

(≈ 20–25 minutes)

Roberts' Theorem

- Similar to GS Theorem, we ask what if the valuations are **unrestricted**, i.e., Θ_i contains all possible valuation functions $\theta_i : A \to \mathbb{R}$, no restriction on the functions is imposed

Roberts' Theorem

- Similar to GS Theorem, we ask what if the valuations are **unrestricted**, i.e., Θ_i contains all possible valuation functions $\theta_i : A \to \mathbb{R}$, no restriction on the functions is imposed
- With this unrestricted space of valuations, we can characterize the class DSIC mechanisms in the quasi-linear domain

Roberts' Theorem

- Similar to GS Theorem, we ask what if the valuations are **unrestricted**, i.e., Θ_i contains all possible valuation functions $\theta_i : A \to \mathbb{R}$, no restriction on the functions is imposed
- With this unrestricted space of valuations, we can characterize the class DSIC mechanisms in the quasi-linear domain
- Roberts' theorem

Roberts' Theorem

- Similar to GS Theorem, we ask what if the valuations are **unrestricted**, i.e., Θ_i contains all possible valuation functions $\theta_i : A \to \mathbb{R}$, no restriction on the functions is imposed
- With this unrestricted space of valuations, we can characterize the class DSIC mechanisms in the quasi-linear domain
- Roberts' theorem

Theorem (Roberts 1979)

Let A be finite with $|A| \geqslant 3$. If the type space is unrestricted, then every ONTO and dominant strategy implementable allocation rule must be an affine maximizer.

Roberts' Theorem

- Similar to GS Theorem, we ask what if the valuations are **unrestricted**, i.e., Θ_i contains all possible valuation functions $\theta_i : A \to \mathbb{R}$, no restriction on the functions is imposed
- With this unrestricted space of valuations, we can characterize the class DSIC mechanisms in the quasi-linear domain
- Roberts' theorem

Theorem (Roberts 1979)

Let A be finite with $|A| \geqslant 3$. If the type space is unrestricted, then every ONTO and dominant strategy implementable allocation rule must be an affine maximizer.

- **Proof reference:** Ron Lavi, Ahuva Mu'alem, and Noam Nisan. "Two simplified proofs for Roberts' theorem." In: Social Choice and Welfare 32 (2009), pp. 407–423.

Roberts' Theorem

- Similar to GS Theorem, we ask what if the valuations are **unrestricted**, i.e., Θ_i contains all possible valuation functions $\theta_i : A \to \mathbb{R}$, no restriction on the functions is imposed
- With this unrestricted space of valuations, we can characterize the class DSIC mechanisms in the quasi-linear domain
- Roberts' theorem

Theorem (Roberts 1979)

Let A be finite with $|A| \geqslant 3$. If the type space is unrestricted, then every ONTO and dominant strategy implementable allocation rule must be an affine maximizer.

- **Proof reference:** Ron Lavi, Ahuva Mu'alem, and Noam Nisan. "Two simplified proofs for Roberts' theorem." In: Social Choice and Welfare 32 (2009), pp. 407–423.
- **Similarity with GS Theorem:** GS Theorem is restricting the class to dictatorships, but here it is restricting to affine maximizers

Money Buys a Richer Class

Without money (GS)

any strict preferences, $|A| \geqslant 3$

With money (quasi-linear, Roberts)

any valuations $\theta_i : A \to \mathbb{R}$, $|A| \geqslant 3$

Money Buys a Richer Class

Without money (GS)

any strict preferences, $|A| \geqslant 3$

all rules

With money (quasi-linear, Roberts)

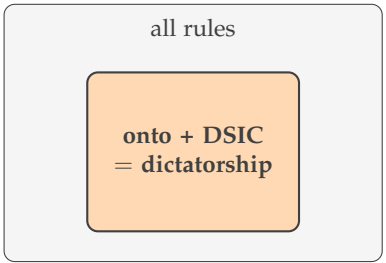
any valuations $\theta_i : A \to \mathbb{R}$, $|A| \geqslant 3$

all rules

Money Buys a Richer Class

Without money (GS)

any strict preferences, $|A| \geqslant 3$



With money (quasi-linear, Roberts)

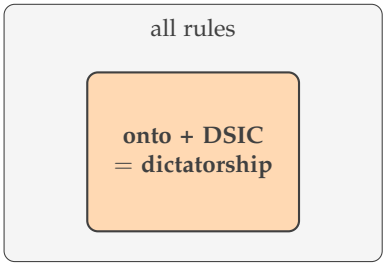
any valuations $\theta_i : A \to \mathbb{R}$, $|A| \geqslant 3$



Money Buys a Richer Class

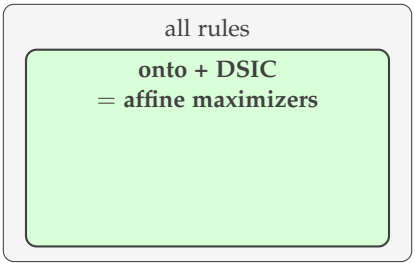
Without money (GS)

any strict preferences, $|A| \geqslant 3$



With money (quasi-linear, Roberts)

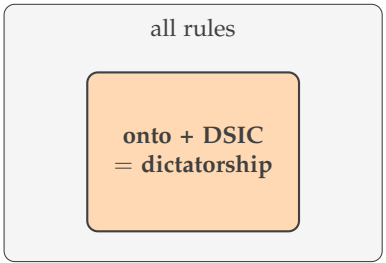
any valuations $\theta_i : A \to \mathbb{R}$, $|A| \geqslant 3$



Money Buys a Richer Class

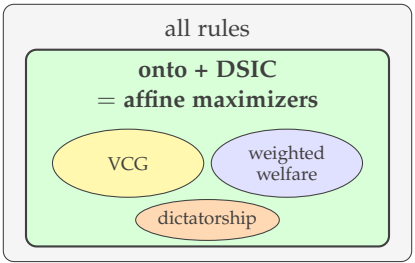
Without money (GS)

any strict preferences, $|A| \geqslant 3$



With money (quasi-linear, Roberts)

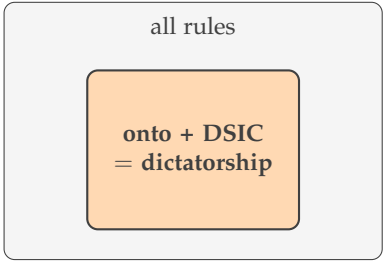
any valuations $\theta_i : A \to \mathbb{R}$, $|A| \geqslant 3$



Money Buys a Richer Class

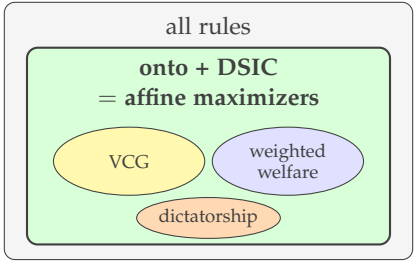
Without money (GS)

any strict preferences, $|A| \geqslant 3$



With money (quasi-linear, Roberts)

any valuations $\theta_i : A \to \mathbb{R}$, $|A| \geqslant 3$



The takeaway

Ordinal and unrestricted $\Rightarrow$ only dictatorships are DSIC. Add money: DSIC = affine maximizers, a much richer class that contains dictatorships, VCG, weighted welfare, and any bonus κ on outcomes.

Escape Route: Restrict the Domain

- Both theorems are bad news for an **unrestricted** domain.

Escape Route: Restrict the Domain

- Both theorems are bad news for an **unrestricted** domain.
- We do not want to be restricted by exactly one rule; also, unrestricted valuations hardly occur in practice.

Escape Route: Restrict the Domain

- Both theorems are bad news for an **unrestricted** domain.
- We do not want to be restricted by exactly one rule; also, unrestricted valuations hardly occur in practice.
- In voting, single-peaked preferences gave us the median voter rule.

Escape Route: Restrict the Domain

- Both theorems are bad news for an **unrestricted** domain.
- We do not want to be restricted by exactly one rule; also, unrestricted valuations hardly occur in practice.
- In voting, single-peaked preferences gave us the median voter rule.
- Here, a natural restriction: **one object, one number per agent**.

Escape Route: Restrict the Domain

- Both theorems are bad news for an **unrestricted** domain.
- We do not want to be restricted by exactly one rule; also, unrestricted valuations hardly occur in practice.
- In voting, single-peaked preferences gave us the median voter rule.
- Here, a natural restriction: **one object, one number per agent**.
- An agent only cares whether she gets the object, and how much she values it.

Escape Route: Restrict the Domain

- Both theorems are bad news for an **unrestricted** domain.
- We do not want to be restricted by exactly one rule; also, unrestricted valuations hardly occur in practice.
- In voting, single-peaked preferences gave us the median voter rule.
- Here, a natural restriction: **one object, one number per agent**.
- An agent only cares whether she gets the object, and how much she values it.
- This domain has truthful mechanisms that are **not** Groves or VCG.

Escape Route: Restrict the Domain

- Both theorems are bad news for an **unrestricted** domain.
- We do not want to be restricted by exactly one rule; also, unrestricted valuations hardly occur in practice.
- In voting, single-peaked preferences gave us the median voter rule.
- Here, a natural restriction: **one object, one number per agent**.
- An agent only cares whether she gets the object, and how much she values it.
- This domain has truthful mechanisms that are **not** Groves or VCG.
- Next: the setup for selling one object.

Setup for selling single indivisible object

- Type set of agent i : $T_i \subseteq \mathbb{R}$

Setup for selling single indivisible object

- Type set of agent i : $T_i \subseteq \mathbb{R}$
- $t_i \in T_i$: value of agent i if she wins the object

Setup for selling single indivisible object

- Type set of agent i : $T_i \subseteq \mathbb{R}$
- $t_i \in T_i$: value of agent i if she wins the object
- An allocation a is a vector of length n that represents the probability of winning the object by the respective agent (a_0 is probability of not selling the object)

$$\textbf{Set of allocations: } \Delta A = \{a \in [0,1]^{n+1} : \sum_{i=0}^{n} a_i = 1\}$$

Setup for selling single indivisible object

- Type set of agent $i : T_i \subseteq \mathbb{R}$
- $t_i \in T_i$: value of agent i if she wins the object
- An allocation a is a vector of length n that represents the probability of winning the object by the respective agent (a_0 is probability of not selling the object)

$$\textbf{Set of allocations: } \Delta A = \{a \in [0,1]^{n+1} : \sum_{i=0}^{n} a_i = 1\}$$

- **Allocation rule:** $f : T_1 \times T_2 \times \ldots \times T_n \to \Delta A$

Setup for selling single indivisible object

- Type set of agent $i : T_i \subseteq \mathbb{R}$
- $t_i \in T_i$: value of agent i if she wins the object
- An allocation a is a vector of length n that represents the probability of winning the object by the respective agent (a_0 is probability of not selling the object)

$$\textbf{Set of allocations: } \Delta A = \{a \in [0,1]^{n+1} : \sum_{i=0}^{n} a_i = 1\}$$

- **Allocation rule:** $f : T_1 \times T_2 \times \ldots \times T_n \to \Delta A$
- **Valuation:** $v_i(a, t_i) = a_i \cdot t_i$ (**product form**, expected valuation)

Setup for selling single indivisible object

- Type set of agent i : $T_i \subseteq \mathbb{R}$
- $t_i \in T_i$: value of agent i if she wins the object
- An allocation a is a vector of length n that represents the probability of winning the object by the respective agent (a_0 is probability of not selling the object)

$$\textbf{Set of allocations: } \Delta A = \{a \in [0,1]^{n+1} : \sum_{i=0}^{n} a_i = 1\}$$

- **Allocation rule:** $f : T_1 \times T_2 \times \ldots \times T_n \to \Delta A$
- **Valuation:** $v_i(a, t_i) = a_i \cdot t_i$ (**product form**, expected valuation)
- Hence, $f_i(t_i, t_{-i})$ is agent i's probability of winning the object when the type profile is (t_i, t_{-i})

Setup for selling single indivisible object

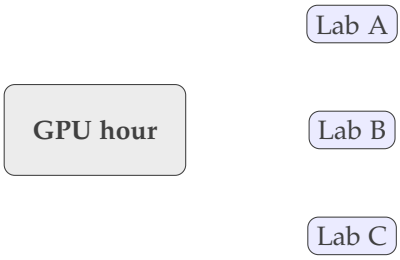
- Type set of agent i : $T_i \subseteq \mathbb{R}$
- $t_i \in T_i$: value of agent i if she wins the object
- An allocation a is a vector of length n that represents the probability of winning the object by the respective agent (a_0 is probability of not selling the object)

$$\textbf{Set of allocations: } \Delta A = \{a \in [0,1]^{n+1} : \sum_{i=0}^{n} a_i = 1\}$$

- **Allocation rule:** $f : T_1 \times T_2 \times \ldots \times T_n \to \Delta A$
- **Valuation:** $v_i(a, t_i) = a_i \cdot t_i$ (**product form**, expected valuation)
- Hence, $f_i(t_i, t_{-i})$ is agent i's probability of winning the object when the type profile is (t_i, t_{-i})
  - $f_0(t)$ is the probability of not selling the object

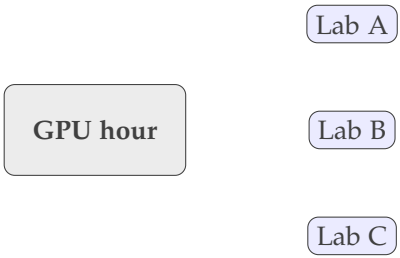
One Number Is Enough

- A cloud provider sells **one GPU hour**.



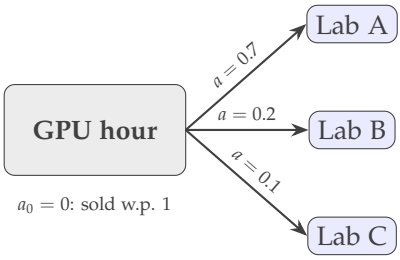
One Number Is Enough

- A cloud provider sells **one GPU hour**.
- Each lab has one number: what the hour is worth to it.



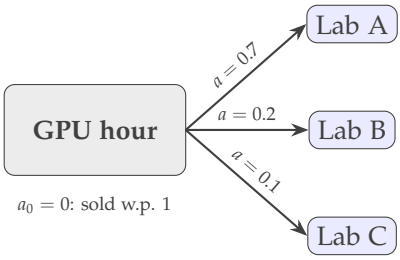
One Number Is Enough

- A cloud provider sells **one GPU hour**.
- Each lab has one number: what the hour is worth to it.
- The provider may even randomize: "you get it with probability 0.7."



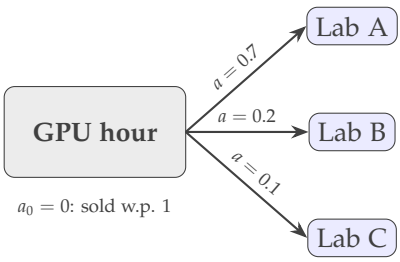
One Number Is Enough

- A cloud provider sells **one GPU hour**.
- Each lab has one number: what the hour is worth to it.
- The provider may even randomize: "you get it with probability 0.7."
- Then the lab's expected value is $0.7 \times t_i$.



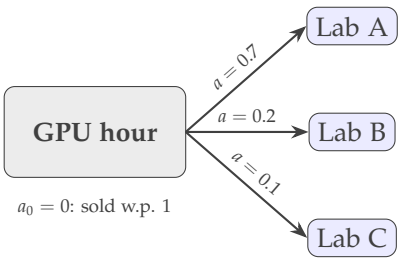
One Number Is Enough

- A cloud provider sells **one GPU hour**.
- Each lab has one number: what the hour is worth to it.
- The provider may even randomize: "you get it with probability 0.7."
- Then the lab's expected value is $0.7 \times t_i$.
- Same story for ad slots: replace a_i by the click-through rate CTR_i.



One Number Is Enough

- A cloud provider sells **one GPU hour**.
- Each lab has one number: what the hour is worth to it.
- The provider may even randomize: "you get it with probability 0.7."
- Then the lab's expected value is $0.7 \times t_i$.
- Same story for ad slots: replace a_i by the click-through rate CTR_i.
- This is the **single parameter domain**.



Vickrey (Second Price) Auction

1. Define $t_{-i}^{(2)} = \max_{j \neq i}\{v_j\}$
2. Agent i wins if $v_i > t_{-i}^{(2)}$, loses if $v_i < t_{-i}^{(2)}$ and a tie breaking rule decides if there is an equality
3. Since payment is $t_{-i}^{(2)}$ if i is the winner, the utility is zero in case of a tie

$$u_i(v_i, v_{-i}) = \begin{cases} 0 & \text{if } v_i \leqslant t_{-i}^{(2)} \\ v_i - t_{-i}^{(2)} & \text{if } v_i > t_{-i}^{(2)} \end{cases}$$

Vickrey (Second Price) Auction

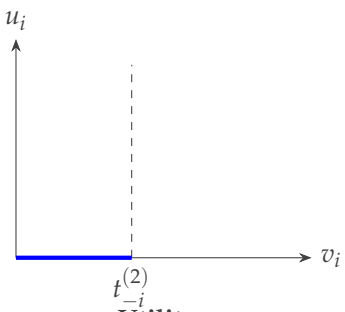
1. Define $t_{-i}^{(2)} = \max_{j \neq i}\{v_j\}$
2. Agent i wins if $v_i > t_{-i}^{(2)}$, loses if $v_i < t_{-i}^{(2)}$ and a tie breaking rule decides if there is an equality
3. Since payment is $t_{-i}^{(2)}$ if i is the winner, the utility is zero in case of a tie

$$u_i(v_i, v_{-i}) = \begin{cases} 0 & \text{if } v_i \leqslant t_{-i}^{(2)} \\ v_i - t_{-i}^{(2)} & \text{if } v_i > t_{-i}^{(2)} \end{cases}$$

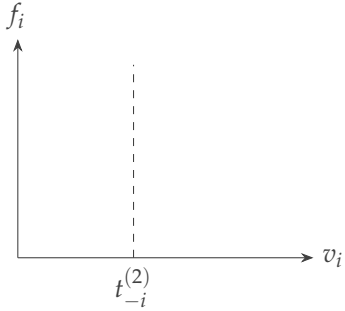
In the GPU-hour auction

Fix the other labs' bids. The best rival bid $t_{-i}^{(2)}$ is a wall. Below the wall you get nothing and pay nothing. Above it you win, and every extra rupee of value is pure profit.

Observations

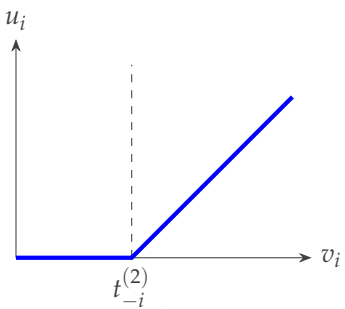


Utility

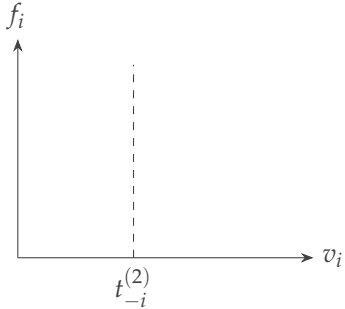


Allocation

Observations

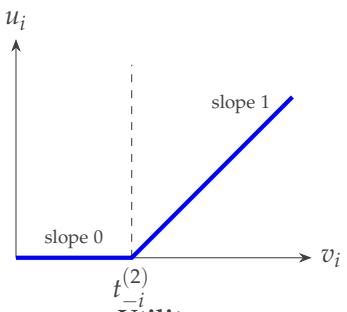


Utility

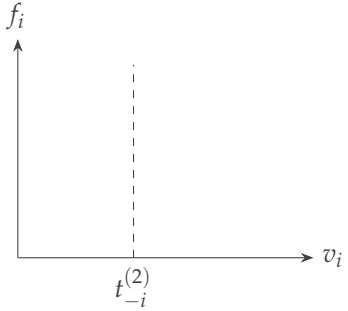


Allocation

Observations



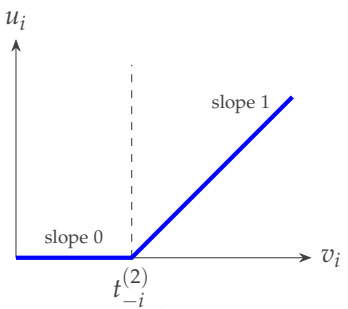
Utility



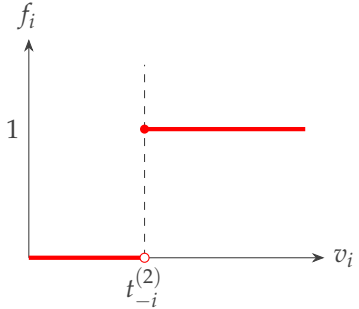
Allocation

- Utility is **convex**, derivative is zero if $v_i < t_{-i}^{(2)}$ and 1 if $v_i > t_{-i}^{(2)}$ (not differentiable at $v_i = t_{-i}^{(2)}$)

Observations



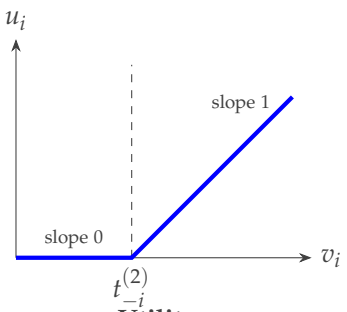
Utility



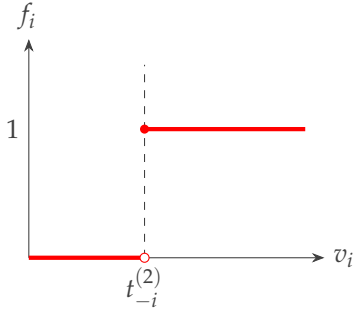
Allocation

- Utility is **convex**, derivative is zero if $v_i < t_{-i}^{(2)}$ and 1 if $v_i > t_{-i}^{(2)}$ (not differentiable at $v_i = t_{-i}^{(2)}$)
- Whenever differentiable, it coincides with the allocation probability

Observations



Utility



Allocation

- Utility is **convex**, derivative is zero if $v_i < t_{-i}^{(2)}$ and 1 if $v_i > t_{-i}^{(2)}$ (not differentiable at $v_i = t_{-i}^{(2)}$)
- Whenever differentiable, it coincides with the allocation probability
- **Hint of the week**: slope of utility = win probability. Hold on to this.



Chunk 4

A Toolkit from Convex Analysis

(≈ 20–25 minutes)

Brief review of convex functions



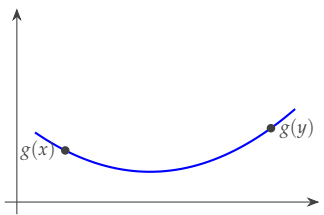
Recall: A function $g : I \to \mathbb{R}$ (where I is an interval) is convex if for every $x, y \in I$ and $\lambda \in [0, 1]$

$$\lambda g(x) + (1 - \lambda) g(y) \geqslant g(\lambda x + (1 - \lambda) y)$$

Brief review of convex functions

Recall: A function $g : I \to \mathbb{R}$ (where I is an interval) is convex if for every $x, y \in I$ and $\lambda \in [0, 1]$

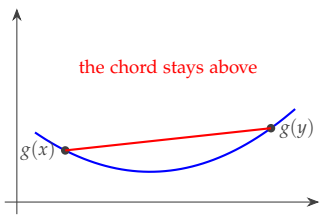
$$\lambda g(x) + (1 - \lambda) g(y) \geqslant g(\lambda x + (1 - \lambda) y)$$



Brief review of convex functions

Recall: A function $g : I \to \mathbb{R}$ (where I is an interval) is convex if for every $x, y \in I$ and $\lambda \in [0, 1]$

$$\lambda g(x) + (1 - \lambda) g(y) \geqslant g(\lambda x + (1 - \lambda) y)$$



Brief review of convex functions (contd.)

Some known facts from convex analysis (see e.g. Rockafellar (1970))

Brief review of convex functions (contd.)

Some known facts from convex analysis (see e.g. Rockafellar (1970))

Fact

*Convex functions are **continuous** in the **interior of its domain***

i.e., jumps can only occur at the boundaries

Brief review of convex functions (contd.)

Some known facts from convex analysis (see e.g. Rockafellar (1970))

Fact

*Convex functions are **continuous** in the **interior of its domain***

i.e., jumps can only occur at the boundaries

Fact

*Convex functions are **differentiable almost everywhere***

i.e., the points where the function is not differentiable form a countable set (see the example before) - has measure zero

Convex functions

If g is differentiable at $x \in I$, we denote the derivative by $g'(x)$

The following definition extends the idea of gradient

Convex functions

If g is differentiable at $x \in I$, we denote the derivative by $g'(x)$

The following definition extends the idea of gradient

Definition (Subgradient)

For any $x \in I$, s^* is a subgradient of g at x if $g(z) \geqslant g(x) + s^*(z - x), \forall z \in I$

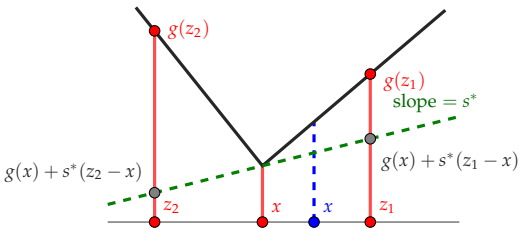
Convex functions

If g is differentiable at $x \in I$, we denote the derivative by $g'(x)$

The following definition extends the idea of gradient

Definition (Subgradient)

For any $x \in I$, s^* is a subgradient of g at x if $g(z) \geqslant g(x) + s^*(z - x), \forall z \in I$



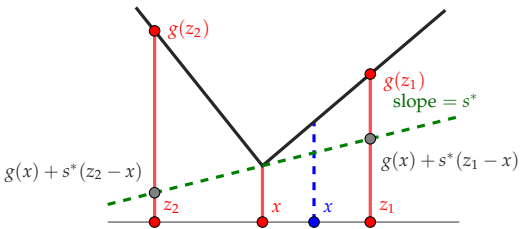
Convex functions

If g is differentiable at $x \in I$, we denote the derivative by $g'(x)$

The following definition extends the idea of gradient

Definition (Subgradient)

For any $x \in I$, s^* is a subgradient of g at x if $g(z) \geqslant g(x) + s^*(z - x), \forall z \in I$



Question

- Always exists?

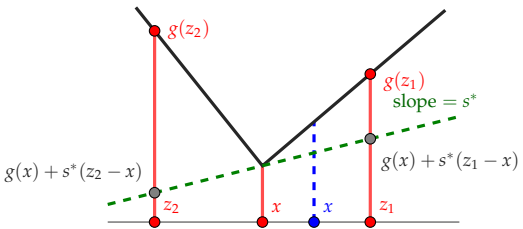
Convex functions

If g is differentiable at $x \in I$, we denote the derivative by $g'(x)$

The following definition extends the idea of gradient

Definition (Subgradient)

For any $x \in I$, s^* is a subgradient of g at x if $g(z) \geqslant g(x) + s^*(z - x), \forall z \in I$

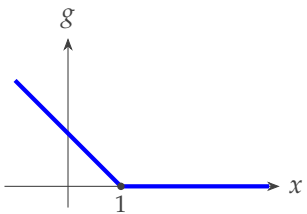


Question

- Always exists?
- Is it unique?

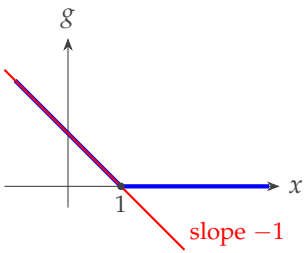
Subgradients of the Hinge Loss

Every ML student has met this kink: the **hinge loss** of an SVM, $g(x) = \max\{0, 1 - x\}$.



Subgradients of the Hinge Loss

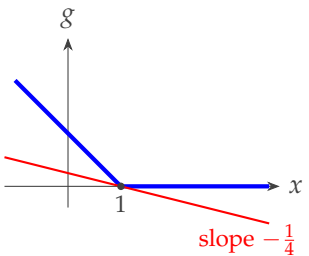
Every ML student has met this kink: the **hinge loss** of an SVM, $g(x) = \max\{0, 1 - x\}$.



- Any line through the kink that stays **below** g is a supporting line.

Subgradients of the Hinge Loss

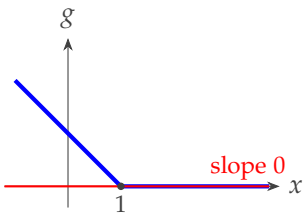
Every ML student has met this kink: the **hinge loss** of an SVM, $g(x) = \max\{0, 1 - x\}$.



- Any line through the kink that stays **below** g is a supporting line.

Subgradients of the Hinge Loss

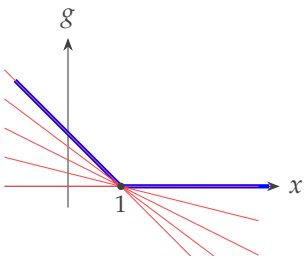
Every ML student has met this kink: the **hinge loss** of an SVM, $g(x) = \max\{0, 1 - x\}$.



- Any line through the kink that stays **below** g is a supporting line.

Subgradients of the Hinge Loss

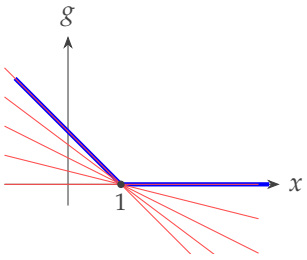
Every ML student has met this kink: the **hinge loss** of an SVM, $g(x) = \max\{0, 1 - x\}$.



- Any line through the kink that stays **below** g is a supporting line.
- Write $\partial g(x)$ for the **set of all subgradients** of g at x. At $x = 1$: $\partial g(1) = [-1, 0]$. Many subgradients. Everywhere else: just one, the slope.

Subgradients of the Hinge Loss

Every ML student has met this kink: the **hinge loss** of an SVM, $g(x) = \max\{0, 1 - x\}$.



- Any line through the kink that stays **below** g is a supporting line.
- Write $\partial g(x)$ for the **set of all subgradients** of g at x. At $x = 1$: $\partial g(1) = [-1, 0]$. Many subgradients. Everywhere else: just one, the slope.
- Optimizers use exactly this in **subgradient descent**.

Standard results

Proofs for the following lemmas can be found in any standard convex analysis text

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Suppose x is in the interior of I and g is differentiable at x. Then $g'(x)$ is the unique subgradient of g.

Standard results

Proofs for the following lemmas can be found in any standard convex analysis text

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Suppose x is in the interior of I and g is differentiable at x. Then $g'(x)$ is the unique subgradient of g.

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Then for every $x \in I$ a subgradient of g at x exists.

Standard results (contd.)

Fact

Let $I' \subseteq I$ be the set of points where g is differentiable. The set $I \setminus I'$ is of measure zero. The set of subgradients at a point forms a convex set.

Standard results (contd.)

Fact

Let $I' \subseteq I$ be the set of points where g is differentiable. The set $I \setminus I'$ is of measure zero. The set of subgradients at a point forms a convex set.

Define $g'_+(x)$ and $g'_-(x)$ as

$$g'_+(x) = \lim_{z \to x,\ z > x} g'(z)$$
$$g'_-(x) = \lim_{z \to x,\ z < x} g'(z)$$

Standard results (contd.)

Fact

Let $I' \subseteq I$ be the set of points where g is differentiable. The set $I \setminus I'$ is of measure zero. The set of subgradients at a point forms a convex set.

Define $g'_+(x)$ and $g'_-(x)$ as

$$g'_+(x) = \lim_{z \to x,\ z > x} g'(z)$$
$$g'_-(x) = \lim_{z \to x,\ z < x} g'(z)$$

Fact

The subgradients at $x \in I \setminus I'$ is $[g'_-(x), g'_+(x)]$

Standard results (contd.)

Fact

Let $I' \subseteq I$ be the set of points where g is differentiable. The set $I \setminus I'$ is of measure zero. The set of subgradients at a point forms a convex set.

Define $g'_+(x)$ and $g'_-(x)$ as

$$g'_+(x) = \lim_{z \to x,\ z > x} g'(z)$$
$$g'_-(x) = \lim_{z \to x,\ z < x} g'(z)$$

Fact

The subgradients at $x \in I \setminus I'$ is $[g'_-(x), g'_+(x)]$

Back to the hinge

At $x = 1$: $g'_- = -1$, $g'_+ = 0$. So $\partial g(1) = [-1, 0]$. The GPU-hour utility at $v_i = t_{-i}^{(2)}$ works the same way.

Summary of the Lemmas

We will denote the set of subgradients of g at $x \in I$ as $\partial g(x)$

Summary of the Lemmas

We will denote the set of subgradients of g at $x \in I$ as $\partial g(x)$

1. First lemma says $\partial g(x) = \{g'(x)\}, \forall x \in I'$

Summary of the Lemmas

We will denote the set of subgradients of g at $x \in I$ as $\partial g(x)$

1. First lemma says $\partial g(x) = \{g'(x)\}, \forall x \in I'$
2. Second lemma says that $\partial g(x) \neq \emptyset, \forall x \in I$

Summary of the Lemmas

We will denote the set of subgradients of g at $x \in I$ as $\partial g(x)$

1. First lemma says $\partial g(x) = \{g'(x)\}, \forall x \in I'$
2. Second lemma says that $\partial g(x) \neq \emptyset, \forall x \in I$

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Let $\phi(z) \in \partial g(z), \forall z \in I$. Then for all $x, y \in I$ such that $x > y$, we have $\phi(x) \geqslant \phi(y)$.

Summary of the Lemmas

We will denote the set of subgradients of g at $x \in I$ as $\partial g(x)$

1. First lemma says $\partial g(x) = \{g'(x)\}, \forall x \in I'$
2. Second lemma says that $\partial g(x) \neq \emptyset, \forall x \in I$

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Let $\phi(z) \in \partial g(z), \forall z \in I$. Then for all $x, y \in I$ such that $x > y$, we have $\phi(x) \geqslant \phi(y)$.

- $\phi(z)$ picks one value at every z (even if subgradients can be many)

Summary of the Lemmas

We will denote the set of subgradients of g at $x \in I$ as $\partial g(x)$

1. First lemma says $\partial g(x) = \{g'(x)\}, \forall x \in I'$
2. Second lemma says that $\partial g(x) \neq \emptyset, \forall x \in I$

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Let $\phi(z) \in \partial g(z), \forall z \in I$. Then for all $x, y \in I$ such that $x > y$, we have $\phi(x) \geqslant \phi(y)$.

- $\phi(z)$ picks one value at every z (even if subgradients can be many)
- This result says that subgradient functions are monotone

Summary of the Lemmas

We will denote the set of subgradients of g at $x \in I$ as $\partial g(x)$

1. First lemma says $\partial g(x) = \{g'(x)\}, \forall x \in I'$
2. Second lemma says that $\partial g(x) \neq \emptyset, \forall x \in I$

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Let $\phi(z) \in \partial g(z), \forall z \in I$. Then for all $x, y \in I$ such that $x > y$, we have $\phi(x) \geqslant \phi(y)$.

- $\phi(z)$ picks one value at every z (even if subgradients can be many)
- This result says that subgradient functions are monotone
- Proof: write $g(x) \geqslant g(y) + \phi(y)(x - y)$ and $g(y) \geqslant g(x) + \phi(x)(y - x)$. Add them: $(\phi(x) - \phi(y))(x - y) \geqslant 0$.

Summary of the Lemmas (contd.)

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Then for any $x, y \in I$

$$g(x) = g(y) + \int_y^x \phi(z)dz$$

where $\phi : I \to \mathbb{R}$ is such that $\phi(z) \in \partial g(z)\ \forall z \in I$

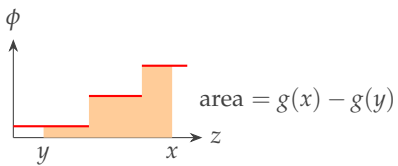
Summary of the Lemmas (contd.)

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Then for any $x, y \in I$

$$g(x) = g(y) + \int_y^x \phi(z)dz$$

where $\phi : I \to \mathbb{R}$ is such that $\phi(z) \in \partial g(z)\ \forall z \in I$



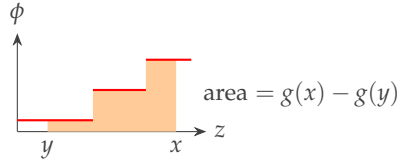
Summary of the Lemmas (contd.)

Lemma

Let $g : I \to \mathbb{R}$ be a convex function. Then for any $x, y \in I$

$$g(x) = g(y) + \int_y^x \phi(z)dz$$

where $\phi : I \to \mathbb{R}$ is such that $\phi(z) \in \partial g(z)\ \forall z \in I$



- Even with kinks, g is recovered by integrating **any** selection of subgradients.
- Next lecture, g is the **utility** and ϕ is the **win probability**.

In-Class Quiz 2

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.

Lecture 21: What We Built



- Affine maximizers: weights w_i for agents, a bonus κ for outcomes. VCG and dictatorship are special cases.

Lecture 21: What We Built

- Affine maximizers: weights w_i for agents, a bonus κ for outcomes. VCG and dictatorship are special cases.
- INA: a zero-weight agent must not break ties. With INA, every AM is DSIC.

Lecture 21: What We Built

- Affine maximizers: weights w_i for agents, a bonus κ for outcomes. VCG and dictatorship are special cases.
- INA: a zero-weight agent must not break ties. With INA, every AM is DSIC.
- Roberts (1979): unrestricted valuations, $|A| \geqslant 3$, onto and DSIC $\Rightarrow$ affine maximizer.

Lecture 21: What We Built

- Affine maximizers: weights w_i for agents, a bonus κ for outcomes. VCG and dictatorship are special cases.
- INA: a zero-weight agent must not break ties. With INA, every AM is DSIC.
- Roberts (1979): unrestricted valuations, $|A| \geqslant 3$, onto and DSIC $\Rightarrow$ affine maximizer.
- Single object, single parameter: valuation $a_i \cdot t_i$. Vickrey's utility is convex, and its slope is the win probability.

Lecture 21: What We Built

- Affine maximizers: weights w_i for agents, a bonus κ for outcomes. VCG and dictatorship are special cases.
- INA: a zero-weight agent must not break ties. With INA, every AM is DSIC.
- Roberts (1979): unrestricted valuations, $|A| \geqslant 3$, onto and DSIC $\Rightarrow$ affine maximizer.
- Single object, single parameter: valuation $a_i \cdot t_i$. Vickrey's utility is convex, and its slope is the win probability.
- Convex toolkit: subgradients exist, are monotone, and integrate back to the function.

Lecture 21: What We Built

- Affine maximizers: weights w_i for agents, a bonus κ for outcomes. VCG and dictatorship are special cases.
- INA: a zero-weight agent must not break ties. With INA, every AM is DSIC.
- Roberts (1979): unrestricted valuations, $|A| \geqslant 3$, onto and DSIC $\Rightarrow$ affine maximizer.
- Single object, single parameter: valuation $a_i \cdot t_i$. Vickrey's utility is convex, and its slope is the win probability.
- Convex toolkit: subgradients exist, are monotone, and integrate back to the function.
- **Next:** Myerson's lemma.



End of Lecture 21

See you next class.

Contents



- Lecture 21: Affine Maximizers and Single Object Allocation
- Lecture 22: Myerson's Lemma and Optimal Mechanism Design

Recap: Two Lemmas We Will Use Today

Let $g : I \to \mathbb{R}$ be convex, and $\phi(z) \in \partial g(z)$ for all $z \in I$.

Lemma

For all $x, y \in I$ with $x > y$: $\phi(x) \geqslant \phi(y)$.

Lemma

For any $x, y \in I$: $g(x) = g(y) + \int_y^x \phi(z)\, dz$.

Recap: Two Lemmas We Will Use Today

Let $g : I \to \mathbb{R}$ be convex, and $\phi(z) \in \partial g(z)$ for all $z \in I$.

Lemma

For all $x, y \in I$ with $x > y$: $\phi(x) \geqslant \phi(y)$.

Lemma

For any $x, y \in I$: $g(x) = g(y) + \int_y^x \phi(z)\, dz$.

The plan for today

Show that DSIC makes the utility convex, with the win probability as its subgradient. Then these two lemmas do all the work.



Chunk 1

Myerson's Lemma: Forward Direction

(≈ 25–30 minutes)

Monotonicity and Myerson's Lemma

Definition

An allocation rule is non-decreasing if for every agent $i \in N$ and $t_{-i} \in T_{-i}$ we have $f_i(t_i, t_{-i}) \geqslant f_i(s_i, t_{-i}),\ \forall s_i, t_i \in T_i,\ t_i > s_i$.

Monotonicity and Myerson's Lemma

Definition

An allocation rule is non-decreasing if for every agent $i \in N$ and $t_{-i} \in T_{-i}$ we have $f_i(t_i, t_{-i}) \geqslant f_i(s_i, t_{-i}),\ \forall s_i, t_i \in T_i,\ t_i > s_i$.

i.e., holding other agents' types fixed, the probability of allocation never decreases with valuation

Monotonicity and Myerson's Lemma

Definition

An allocation rule is non-decreasing if for every agent $i \in N$ and $t_{-i} \in T_{-i}$ we have $f_i(t_i, t_{-i}) \geqslant f_i(s_i, t_{-i}),\ \forall s_i, t_i \in T_i,\ t_i > s_i$.

i.e., holding other agents' types fixed, the probability of allocation never decreases with valuation

Theorem (Myerson 1981)

Suppose $T_i = [0, b_i],\ \forall i \in N$, and the valuations are in the product form. An allocation rule $f : T \to \Delta A$ and a payment rule $(p_1, p_2, \ldots, p_n)$ are DSIC iff

Monotonicity and Myerson's Lemma

Definition

An allocation rule is non-decreasing if for every agent $i \in N$ and $t_{-i} \in T_{-i}$ we have $f_i(t_i, t_{-i}) \geqslant f_i(s_i, t_{-i}),\ \forall s_i, t_i \in T_i,\ t_i > s_i$.

i.e., holding other agents' types fixed, the probability of allocation never decreases with valuation

Theorem (Myerson 1981)

Suppose $T_i = [0, b_i],\ \forall i \in N$, and the valuations are in the product form. An allocation rule $f : T \to \Delta A$ and a payment rule $(p_1, p_2, \ldots, p_n)$ are DSIC iff

- *f is non-decreasing, and*

Monotonicity and Myerson's Lemma

Definition

An allocation rule is non-decreasing if for every agent $i \in N$ and $t_{-i} \in T_{-i}$ we have $f_i(t_i, t_{-i}) \geqslant f_i(s_i, t_{-i}),\ \forall s_i, t_i \in T_i,\ t_i > s_i$.

i.e., holding other agents' types fixed, the probability of allocation never decreases with valuation

Theorem (Myerson 1981)

Suppose $T_i = [0, b_i],\ \forall i \in N$, and the valuations are in the product form. An allocation rule $f : T \to \Delta A$ and a payment rule $(p_1, p_2, \ldots, p_n)$ are DSIC iff

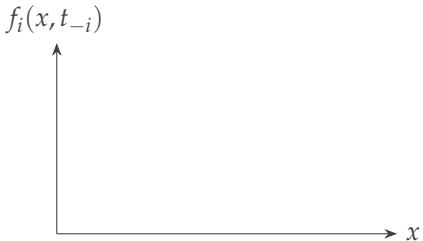
- *f is non-decreasing, and*
- *payments are given by*

$$p_i(t_i, t_{-i}) = p_i(0, t_{-i}) + t_i f_i(t_i, t_{-i}) - \int_0^{t_i} f_i(x, t_{-i})\, dx, \quad \forall t_i \in T_i,\ \forall t_{-i} \in T_{-i},\ \forall i \in N$$

The Spot-GPU Lottery

$$p_i(t_i, t_{-i}) = \underbrace{p_i(0, t_{-i})}_{\text{constant } (=0 \text{ here})} + \underbrace{t_i f_i(t_i, t_{-i})}_{\text{the rectangle}} - \underbrace{\int_0^{t_i} f_i(x, t_{-i})\, dx}_{\text{orange area}}$$

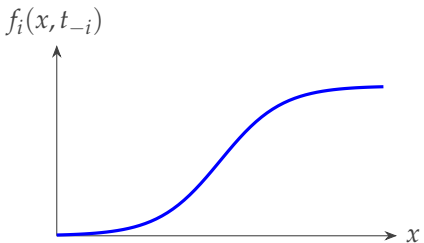
- A cloud provider has a spare GPU. It runs a **lottery** weighted by bids.



The Spot-GPU Lottery

$$p_i(t_i, t_{-i}) = \underbrace{p_i(0, t_{-i})}_{\text{constant } (=0 \text{ here})} + \underbrace{t_i f_i(t_i, t_{-i})}_{\text{the rectangle}} - \underbrace{\int_0^{t_i} f_i(x, t_{-i})\, dx}_{\text{orange area}}$$

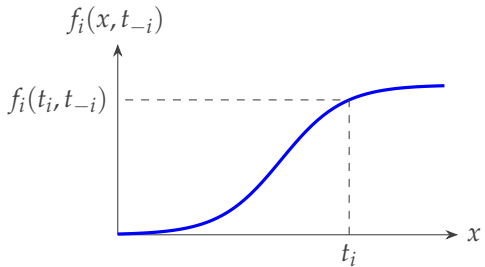
- A cloud provider has a spare GPU. It runs a **lottery** weighted by bids.
- Fix everyone else's bid. Your win chance rises with your bid. That is the blue curve.



The Spot-GPU Lottery

$$p_i(t_i, t_{-i}) = \underbrace{p_i(0, t_{-i})}_{\text{constant } (=0 \text{ here})} + \underbrace{t_i f_i(t_i, t_{-i})}_{\text{the rectangle}} - \underbrace{\int_0^{t_i} f_i(x, t_{-i})\, dx}_{\text{orange area}}$$

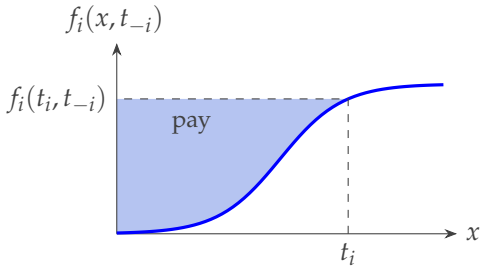
- A cloud provider has a spare GPU. It runs a **lottery** weighted by bids.
- Fix everyone else's bid. Your win chance rises with your bid. That is the blue curve.
- Myerson: the rectangle $t_i f_i(t_i, t_{-i})$ splits into two parts.



The Spot-GPU Lottery

$$p_i(t_i, t_{-i}) = \underbrace{p_i(0, t_{-i})}_{\text{constant } (=0 \text{ here})} + \underbrace{t_i f_i(t_i, t_{-i})}_{\text{the rectangle}} - \underbrace{\int_0^{t_i} f_i(x, t_{-i})\, dx}_{\text{orange area}}$$

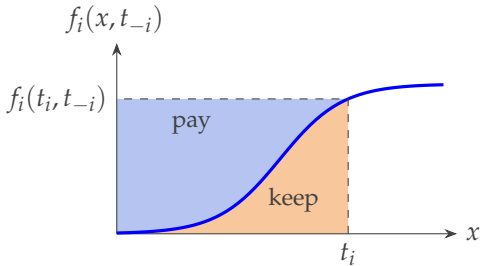
- A cloud provider has a spare GPU. It runs a **lottery** weighted by bids.
- Fix everyone else's bid. Your win chance rises with your bid. That is the blue curve.
- Myerson: the rectangle $t_i f_i(t_i, t_{-i})$ splits into two parts.
- **Blue**: what you **pay**.



The Spot-GPU Lottery

$$p_i(t_i, t_{-i}) = \underbrace{p_i(0, t_{-i})}_{\text{constant } (=0 \text{ here})} + \underbrace{t_i f_i(t_i, t_{-i})}_{\text{the rectangle}} - \underbrace{\int_0^{t_i} f_i(x, t_{-i})\, dx}_{\text{orange area}}$$

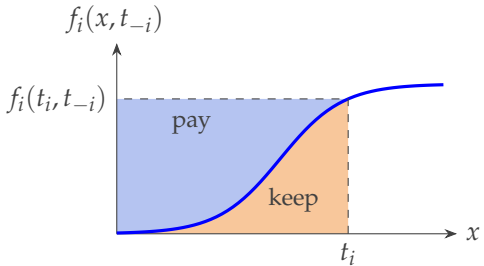
- A cloud provider has a spare GPU. It runs a **lottery** weighted by bids.
- Fix everyone else's bid. Your win chance rises with your bid. That is the blue curve.
- Myerson: the rectangle $t_i f_i(t_i, t_{-i})$ splits into two parts.
- **Blue**: what you **pay**.
- **Orange**: what you **keep**.



The Spot-GPU Lottery

$$p_i(t_i, t_{-i}) = \underbrace{p_i(0, t_{-i})}_{\text{constant } (=0 \text{ here})} + \underbrace{t_i f_i(t_i, t_{-i})}_{\text{the rectangle}} - \underbrace{\int_0^{t_i} f_i(x, t_{-i})\, dx}_{\text{orange area}}$$

- A cloud provider has a spare GPU. It runs a **lottery** weighted by bids.
- Fix everyone else's bid. Your win chance rises with your bid. That is the blue curve.
- Myerson: the rectangle $t_i f_i(t_i, t_{-i})$ splits into two parts.
- **Blue**: what you **pay**.
- **Orange**: what you **keep**.
- The curve alone fixes the price. No freedom, except the constant $p_i(0, t_{-i})$ ($= 0$ for simplicity).



drawn with $p_i(0, t_{-i}) = 0$

Proof of Myerson's Lemma



Remark: Difference with Roberts' theorem

Roberts' result gives a functional form, while Myerson's result is a more implicit property. Sometimes functional forms help answering questions in a more direct manner.

Proof of Myerson's Lemma

Remark: Difference with Roberts' theorem

Roberts' result gives a functional form, while Myerson's result is a more implicit property. Sometimes functional forms help answering questions in a more direct manner.

Proof: Forward direction

- Given (f, p) is DSIC

Proof of Myerson's Lemma

Remark: Difference with Roberts' theorem

Roberts' result gives a functional form, while Myerson's result is a more implicit property. Sometimes functional forms help answering questions in a more direct manner.

Proof: Forward direction

- Given (f, p) is DSIC
- Utility of agent i at types t_i and s_i respectively:

$$u_i(t_i, t_{-i}) = t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}), \text{ and } u_i(s_i, t_{-i}) = s_i f_i(s_i, t_{-i}) - p_i(s_i, t_{-i})$$

Proof of Myerson's Lemma

Remark: Difference with Roberts' theorem

Roberts' result gives a functional form, while Myerson's result is a more implicit property. Sometimes functional forms help answering questions in a more direct manner.

Proof: Forward direction

- Given (f, p) is DSIC
- Utility of agent i at types t_i and s_i respectively:

$$u_i(t_i, t_{-i}) = t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}), \text{ and } u_i(s_i, t_{-i}) = s_i f_i(s_i, t_{-i}) - p_i(s_i, t_{-i})$$

- Since (f, p) is DSIC, we have $\forall t_i, s_i, t_{-i}$

$$\begin{aligned} u_i(t_i, t_{-i}) = t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) &\geqslant t_i f_i(s_i, t_{-i}) - p_i(s_i, t_{-i}) \\ &= s_i f_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i}) - p_i(s_i, t_{-i}) \\ &= u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i}) \end{aligned}$$

Proof of Myerson's Lemma (contd.)



Proof: Forward direction

- We have, $u_i(t_i, t_{-i}) \geqslant u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i})$

Proof of Myerson's Lemma (contd.)



Proof: Forward direction

- We have, $u_i(t_i, t_{-i}) \geqslant u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i})$
- Fixing t_{-i}, define $g(t_i) = u_i(t_i, t_{-i})$, $\phi(t_i) = f_i(t_i, t_{-i})$

Proof of Myerson's Lemma (contd.)



Proof: Forward direction

- We have, $u_i(t_i, t_{-i}) \geqslant u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i})$
- Fixing t_{-i}, define $g(t_i) = u_i(t_i, t_{-i})$, $\phi(t_i) = f_i(t_i, t_{-i})$
- Hence, the above inequality can be written as

$$g(t_i) \geqslant g(s_i) + \phi(s_i)(t_i - s_i)$$

Proof of Myerson's Lemma (contd.)



Proof: Forward direction

- We have, $u_i(t_i, t_{-i}) \geqslant u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i})$
- Fixing t_{-i}, define $g(t_i) = u_i(t_i, t_{-i})$, $\phi(t_i) = f_i(t_i, t_{-i})$
- Hence, the above inequality can be written as

$$g(t_i) \geqslant g(s_i) + \phi(s_i)(t_i - s_i)$$

- The above implies $\phi(s_i)$ is a sub-gradient of g at s_i, if g is convex

Proof of Myerson's Lemma (contd.)

Proof: Forward direction

- We have, $u_i(t_i, t_{-i}) \geqslant u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i})$
- Fixing t_{-i}, define $g(t_i) = u_i(t_i, t_{-i})$, $\phi(t_i) = f_i(t_i, t_{-i})$
- Hence, the above inequality can be written as

$$g(t_i) \geqslant g(s_i) + \phi(s_i)(t_i - s_i)$$

- The above implies $\phi(s_i)$ is a sub-gradient of g at s_i, if g is convex
- Need to show that g is convex

Proof of Myerson's Lemma (contd.)

Proof: Forward direction

- We have, $u_i(t_i, t_{-i}) \geqslant u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i})$
- Fixing t_{-i}, define $g(t_i) = u_i(t_i, t_{-i})$, $\phi(t_i) = f_i(t_i, t_{-i})$
- Hence, the above inequality can be written as

$$g(t_i) \geqslant g(s_i) + \phi(s_i)(t_i - s_i)$$

- The above implies $\phi(s_i)$ is a sub-gradient of g at s_i, if g is convex
- Need to show that g is convex
- Pick $x_i, z_i \in T_i$ and define $y_i = \lambda x_i + (1 - \lambda) z_i$, where $\lambda \in [0, 1]$

Proof of Myerson's Lemma (contd.)

Proof: Forward direction

- We have, $u_i(t_i, t_{-i}) \geqslant u_i(s_i, t_{-i}) + (t_i - s_i) f_i(s_i, t_{-i})$
- Fixing t_{-i}, define $g(t_i) = u_i(t_i, t_{-i})$, $\phi(t_i) = f_i(t_i, t_{-i})$
- Hence, the above inequality can be written as

$$g(t_i) \geqslant g(s_i) + \phi(s_i)(t_i - s_i)$$

- The above implies $\phi(s_i)$ is a sub-gradient of g at s_i, if g is convex
- Need to show that g is convex
- Pick $x_i, z_i \in T_i$ and define $y_i = \lambda x_i + (1 - \lambda) z_i$, where $\lambda \in [0, 1]$
- DSIC implies

$$g(x_i) \geqslant g(y_i) + \phi(y_i)(x_i - y_i) \text{ and } g(z_i) \geqslant g(y_i) + \phi(y_i)(z_i - y_i)$$

Proof of Myerson's Lemma (contd.)

Proof: Forward direction

$$g(x_i) \geqslant g(y_i) + \phi(y_i)(x_i - y_i) \text{ and } g(z_i) \geqslant g(y_i) + \phi(y_i)(z_i - y_i)$$
$$\implies \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(y_i) + \phi(y_i)(\lambda x_i + (1-\lambda) z_i - y_i)$$
$$\implies \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(\lambda x_i + (1-\lambda) z_i)$$

- Thus, g is convex

Proof of Myerson's Lemma (contd.)

Proof: Forward direction

$$
\begin{aligned}
& g(x_i) \geqslant g(y_i) + \phi(y_i)(x_i - y_i) \text{ and } g(z_i) \geqslant g(y_i) + \phi(y_i)(z_i - y_i) \\
\implies & \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(y_i) + \phi(y_i)(\lambda x_i + (1-\lambda) z_i - y_i) \\
\implies & \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(\lambda x_i + (1-\lambda) z_i)
\end{aligned}
$$

- Thus, g is convex
- Apply the two lemmas from our review of convex analysis

Proof of Myerson's Lemma (contd.)

Proof: Forward direction

$$g(x_i) \geqslant g(y_i) + \phi(y_i)(x_i - y_i) \text{ and } g(z_i) \geqslant g(y_i) + \phi(y_i)(z_i - y_i)$$
$$\implies \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(y_i) + \phi(y_i)(\lambda x_i + (1-\lambda) z_i - y_i)$$
$$\implies \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(\lambda x_i + (1-\lambda) z_i)$$

- Thus, g is convex
- Apply the two lemmas from our review of convex analysis
- Monotone subgradients $\implies$ $\phi = f_i(., t_{-i})$ is non-decreasing $\implies$ **part 1 proved**

Proof of Myerson's Lemma (contd.)

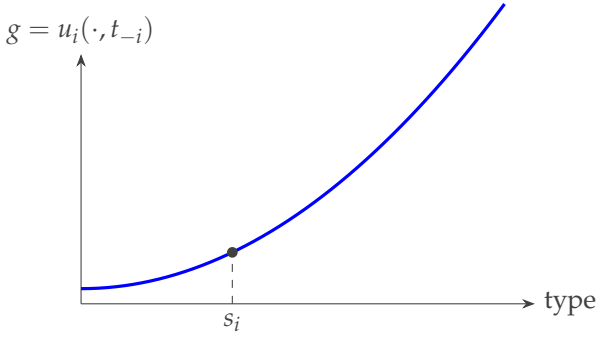
Proof: Forward direction

$$g(x_i) \geqslant g(y_i) + \phi(y_i)(x_i - y_i) \text{ and } g(z_i) \geqslant g(y_i) + \phi(y_i)(z_i - y_i)$$
$$\implies \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(y_i) + \phi(y_i)(\lambda x_i + (1-\lambda) z_i - y_i)$$
$$\implies \lambda g(x_i) + (1-\lambda) g(z_i) \geqslant g(\lambda x_i + (1-\lambda) z_i)$$

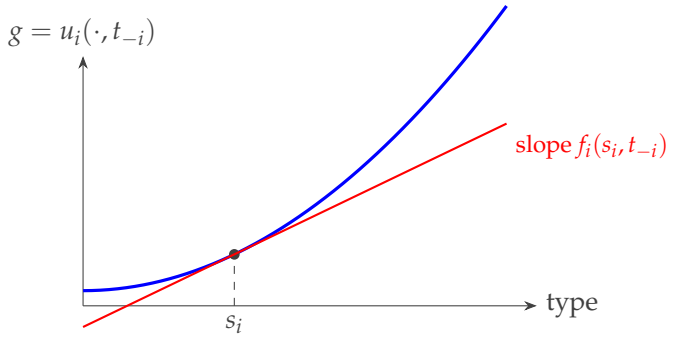
- Thus, g is convex
- Apply the two lemmas from our review of convex analysis
- Monotone subgradients $\implies$ $\phi = f_i(., t_{-i})$ is non-decreasing $\implies$ **part 1 proved**
- Integral lemma $\implies$

$$g(t_i) = g(0) + \int_0^{t_i} \phi(x) dx \implies u_i(t_i, t_{-i}) = u_i(0, t_{-i}) + \int_0^{t_i} f_i(x, t_{-i}) dx$$
$$\implies t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) = -p_i(0, t_{-i}) + \int_0^{t_i} f_i(x, t_{-i}) dx$$
$$\implies p_i(t_i, t_{-i}) = p_i(0, t_{-i}) + t_i f_i(t_i, t_{-i}) - \int_0^{t_i} f_i(x, t_{-i}) dx \implies \textbf{part 2 proved}$$

The Inequality, Drawn

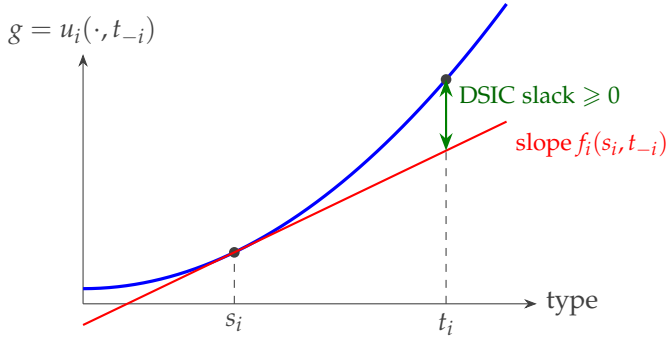


The Inequality, Drawn



- The line through $(s_i, g(s_i))$ with slope $f_i(s_i, t_{-i})$ stays **below** g everywhere.

The Inequality, Drawn



- The line through $(s_i, g(s_i))$ with slope $f_i(s_i, t_{-i})$ stays **below** g everywhere.
- That is the subgradient picture from last lecture. The win probability is the slope.

Chunk 2

Reverse Direction and Examples

(≈ 20–25 minutes)

Proof of Myerson's Lemma (contd.)



Proof: Reverse direction

- Given f is non-decreasing and the payment formula

Proof of Myerson's Lemma (contd.)



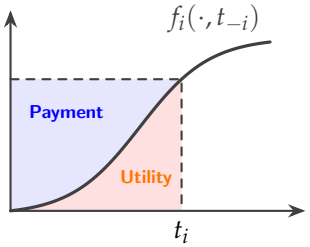
Proof: Reverse direction

- Given f is non-decreasing and the payment formula
- Proof by pictures: assume $p_i(0, t_{-i}) = 0$

Proof of Myerson's Lemma (contd.)

Proof: Reverse direction

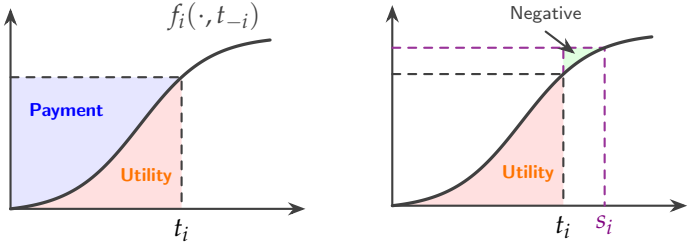
- Given f is non-decreasing and the payment formula
- Proof by pictures: assume $p_i(0, t_{-i}) = 0$



Proof of Myerson's Lemma (contd.)

Proof: Reverse direction

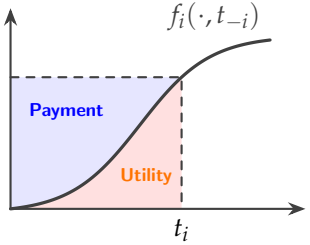
- Given f is non-decreasing and the payment formula
- Proof by pictures: assume $p_i(0, t_{-i}) = 0$

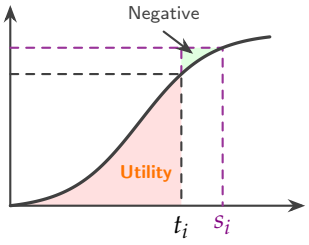


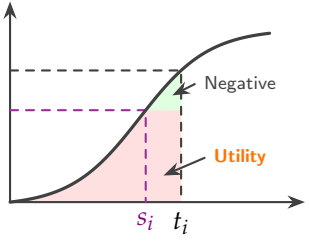
Proof of Myerson's Lemma (contd.)

Proof: Reverse direction

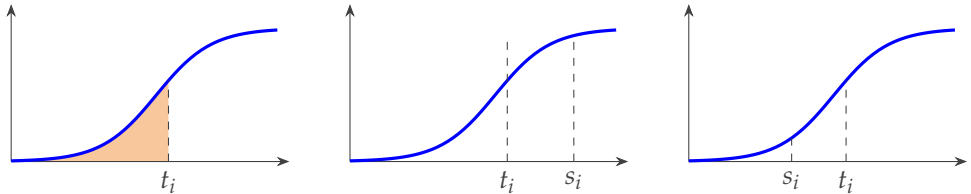
- Given f is non-decreasing and the payment formula
- Proof by pictures: assume $p_i(0, t_{-i}) = 0$





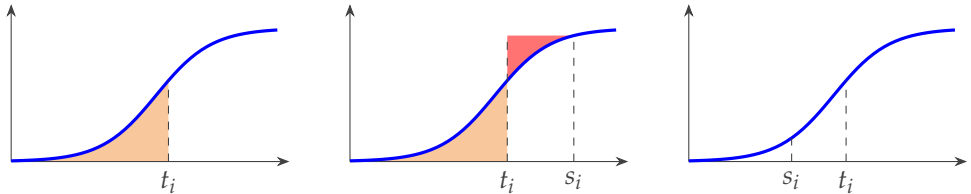


The Shady Bid at the Spot-GPU Lottery



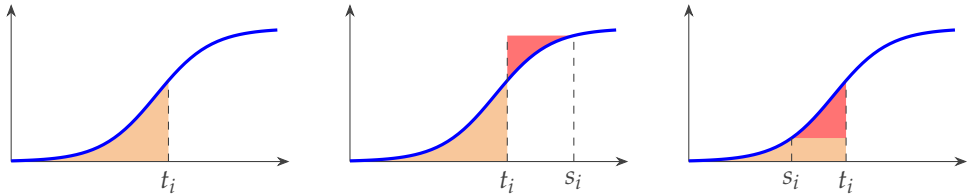
- **Orange**: what you keep. **Red**: what you lose by lying.

The Shady Bid at the Spot-GPU Lottery



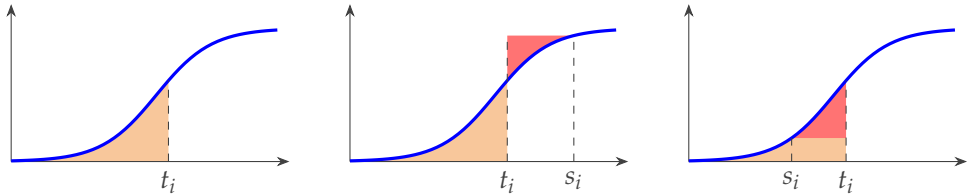
- **Orange**: what you keep. **Red**: what you lose by lying.
- Overbid: you pay for win chances you value **less** than their price.

The Shady Bid at the Spot-GPU Lottery



- **Orange**: what you keep. **Red**: what you lose by lying.
- Overbid: you pay for win chances you value **less** than their price.
- Underbid: you give up win chances that were **cheap** for you.

The Shady Bid at the Spot-GPU Lottery



- **Orange**: what you keep. **Red**: what you lose by lying.
- Overbid: you pay for win chances you value **less** than their price.
- Underbid: you give up win chances that were **cheap** for you.
- The red region exists only because the curve goes **up**. Monotone curve $\Rightarrow$ red $\geqslant 0$.

Proof of Myerson's Lemma (contd.)

Proof: Reverse direction

- Given f is non-decreasing and the payment formula

Proof of Myerson's Lemma (contd.)

Proof: Reverse direction

- Given f is non-decreasing and the payment formula
- Proof by pictures: assume $p_i(0, t_{-i}) = 0$

$$[t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i})] - [t_i f_i(s_i, t_{-i}) - p_i(s_i, t_{-i})] = (s_i - t_i) f_i(s_i, t_{-i}) + \int_{s_i}^{t_i} f_i(x, t_{-i})\, dx \geqslant 0$$

Proof of Myerson's Lemma (contd.)

Proof: Reverse direction

- Given f is non-decreasing and the payment formula
- Proof by pictures: assume $p_i(0, t_{-i}) = 0$

$$[t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i})] - [t_i f_i(s_i, t_{-i}) - p_i(s_i, t_{-i})] = (s_i - t_i) f_i(s_i, t_{-i}) + \int_{s_i}^{t_i} f_i(x, t_{-i})\, dx \geqslant 0$$

Corollary

An allocation rule in a single object allocation setting is implementable in dominant strategies iff it is non-decreasing.

Proof of Myerson's Lemma (contd.)

Proof: Reverse direction

- Given f is non-decreasing and the payment formula
- Proof by pictures: assume $p_i(0, t_{-i}) = 0$

$$[t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i})] - [t_i f_i(s_i, t_{-i}) - p_i(s_i, t_{-i})] = (s_i - t_i) f_i(s_i, t_{-i}) + \int_{s_i}^{t_i} f_i(x, t_{-i})\, dx \geqslant 0$$

Corollary

An allocation rule in a single object allocation setting is implementable in dominant strategies iff it is non-decreasing.

Why a bidding bot cares

Train an agent to bid against a Myerson mechanism. Its best policy is the identity map: bid your value. No opponent model is needed. Cognitive load is minimal.

Examples of single object allocation

1. Constant allocation rule - non-decreasing, payment = constant (e.g. 0)
   A hackathon gives the GPU to a random team, whatever they bid.

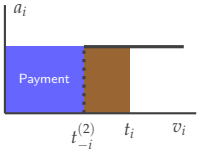
Examples of single object allocation

1. Constant allocation rule - non-decreasing, payment = constant (e.g. 0)
   A hackathon gives the GPU to a random team, whatever they bid.
2. Dictatorial - give the object only to the dictator - non decreasing, payment = constant / zero
   The founder's pet project always gets the GPU.

Examples of single object allocation

1. Constant allocation rule - non-decreasing, payment = constant (e.g. 0)
   A hackathon gives the GPU to a random team, whatever they bid.
2. Dictatorial - give the object only to the dictator - non decreasing, payment = constant / zero
   The founder's pet project always gets the GPU.
3. Second price auction

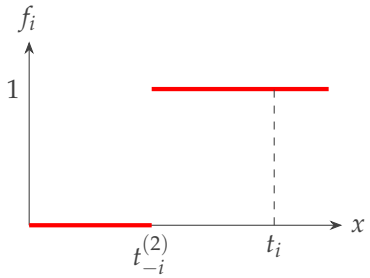
$$p_i(0, t_{-i}) + t_i f_i(t_i, t_{-i}) - \int_0^{t_i} f_i(x, t_{-i}) dx$$



Allocation for second price auction

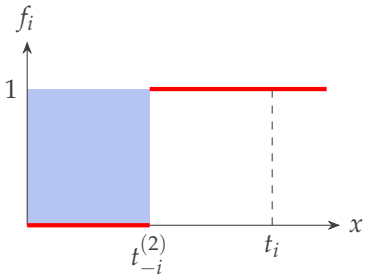
The Second Price Payment, Computed

- The curve is a step at $t_{-i}^{(2)}$.



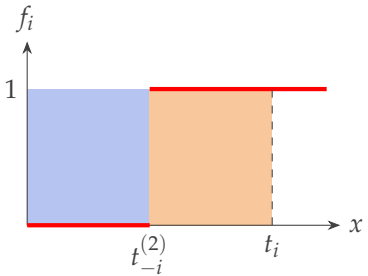
The Second Price Payment, Computed

- The curve is a step at $t_{-i}^{(2)}$.
- Rectangle: $t_i \times 1 = t_i$.



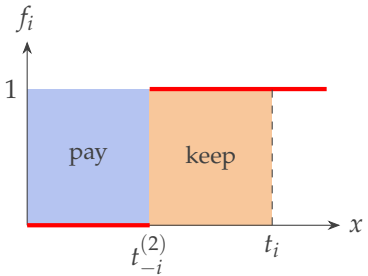
The Second Price Payment, Computed

- The curve is a step at $t_{-i}^{(2)}$.
- Rectangle: $t_i \times 1 = t_i$.
- Area under the step: $t_i - t_{-i}^{(2)}$.



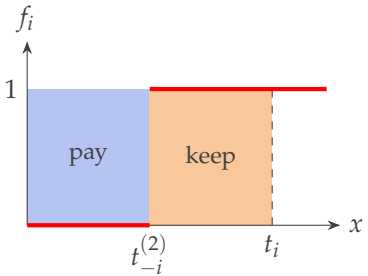
The Second Price Payment, Computed

- The curve is a step at $t_{-i}^{(2)}$.
- Rectangle: $t_i \times 1 = t_i$.
- Area under the step: $t_i - t_{-i}^{(2)}$.
- Payment $= 0 + t_i - (t_i - t_{-i}^{(2)}) = t_{-i}^{(2)}$.



The Second Price Payment, Computed

- The curve is a step at $t_{-i}^{(2)}$.
- Rectangle: $t_i \times 1 = t_i$.
- Area under the step: $t_i - t_{-i}^{(2)}$.
- Payment $= 0 + t_i - (t_i - t_{-i}^{(2)}) = t_{-i}^{(2)}$.
- Myerson **derives** Vickrey's price.



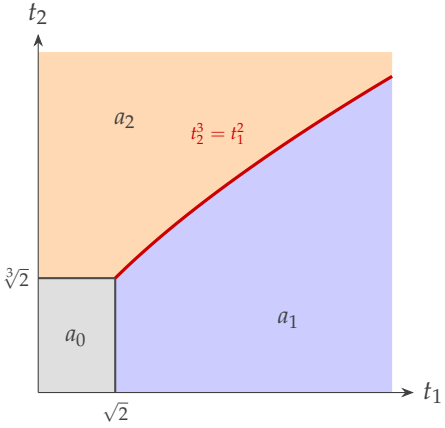
Examples of single object allocation

4. Efficient allocation with a reserve price is also non decreasing. If the highest value is below a reserve price r, nobody gets the object. Otherwise, the item goes to the highest bidder.
   Allocated to i if $v_i > \max\{t_{-i}^{(2)}, r\}$. Payment = $\max\{t_{-i}^{(2)}, r\}$
   An auction with a floor price: no sale below the floor.

Examples of single object allocation

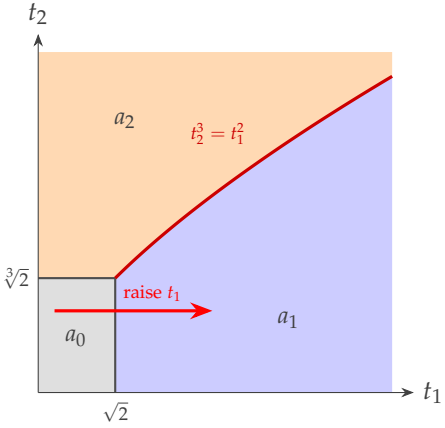
4. Efficient allocation with a reserve price is also non decreasing. If the highest value is below a reserve price r, nobody gets the object. Otherwise, the item goes to the highest bidder.
   Allocated to i if $v_i > \max\{t_{-i}^{(2)}, r\}$. Payment = $\max\{t_{-i}^{(2)}, r\}$
   An auction with a floor price: no sale below the floor.
5. Not so common allocation rule: $N = \{1, 2\}$, $A = \{a_0, a_1, a_2\}$ Given a type profile $t = (t_1, t_2)$, the seller computes $m(t) = \max\{2, t_1^2, t_2^3\}$ - select a_0, a_1, a_2 depending on which of the three expressions is the maxima - break ties in favor of $0 > 1 > 2$
   Player 1 gets the object if $t_1 > \sqrt{\max\{2, t_2^3\}}$
   Player 2 gets the object if $t_2 > \sqrt[3]{\max\{2, t_1^2\}}$

The Odd Seller, Drawn



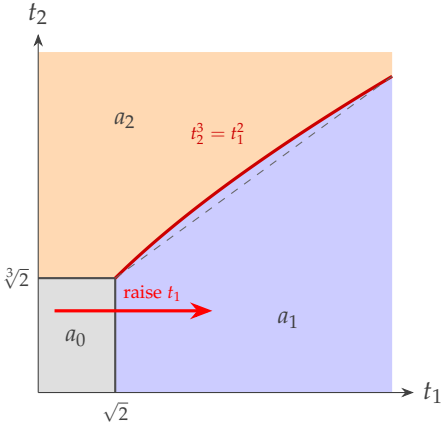
- Gray: the seller keeps the object.

The Odd Seller, Drawn



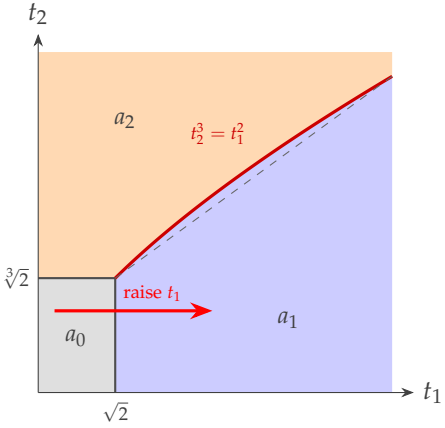
- Gray: the seller keeps the object.
- Fix t_2 and slide t_1 to the right. Once you enter blue, you **stay** in blue.

The Odd Seller, Drawn



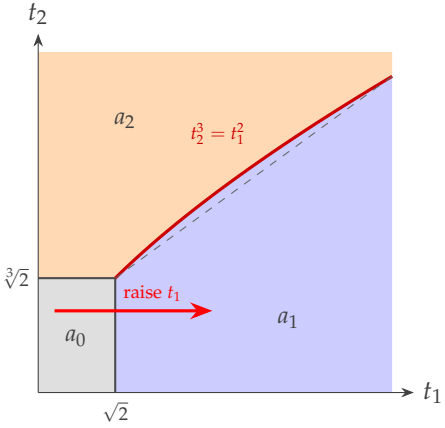
- Gray: the seller keeps the object.
- Fix t_2 and slide t_1 to the right. Once you enter blue, you **stay** in blue.
- The blue–orange boundary $t_2 = t_1^{2/3}$ is **curved**. The other two are straight: $t_1 = \sqrt{2}$ and $t_2 = \sqrt[3]{2}$.

The Odd Seller, Drawn



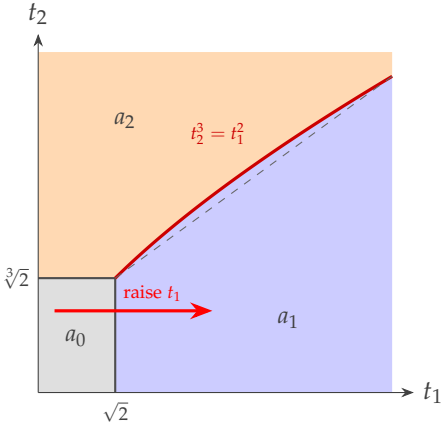
- Gray: the seller keeps the object.
- Fix t_2 and slide t_1 to the right. Once you enter blue, you **stay** in blue.
- The blue–orange boundary $t_2 = t_1^{2/3}$ is **curved**. The other two are straight: $t_1 = \sqrt{2}$ and $t_2 = \sqrt[3]{2}$.
- So $f_1(\cdot, t_2)$ is a step: non-decreasing. Same for agent 2 going up.

The Odd Seller, Drawn



- Gray: the seller keeps the object.
- Fix t_2 and slide t_1 to the right. Once you enter blue, you **stay** in blue.
- The blue–orange boundary $t_2 = t_1^{2/3}$ is **curved**. The other two are straight: $t_1 = \sqrt{2}$ and $t_2 = \sqrt[3]{2}$.
- So $f_1(\cdot, t_2)$ is a step: non-decreasing. Same for agent 2 going up.
- By the corollary, it is DSIC with a suitable payment.

The Odd Seller, Drawn



- Gray: the seller keeps the object.
- Fix t_2 and slide t_1 to the right. Once you enter blue, you **stay** in blue.
- The blue–orange boundary $t_2 = t_1^{2/3}$ is **curved**. The other two are straight: $t_1 = \sqrt{2}$ and $t_2 = \sqrt[3]{2}$.
- So $f_1(\cdot, t_2)$ is a step: non-decreasing. Same for agent 2 going up.
- By the corollary, it is DSIC with a suitable payment.
- Payment of the winner: the **threshold** where her step jumps.



In-Class Quiz 1

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.

Chunk 3

Individual Rationality and Non-Vickrey Auctions

(≈ 20–25 minutes)

Individual Rationality

Definition

A mechanism (f, p) is **ex-post individually rational** if

$$t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) \geqslant 0, \ \forall t_i \in T_i, t_{-i} \in T_{-i}, \forall i \in N$$

Individual Rationality

Definition

A mechanism (f, p) is **ex-post individually rational** if

$$t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) \geqslant 0, \ \forall t_i \in T_i, t_{-i} \in T_{-i}, \forall i \in N$$

Ex-post: Even after all agents have revealed their types, participating is weakly preferred.

Individual Rationality

Definition

A mechanism (f, p) is **ex-post individually rational** if

$$t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) \geqslant 0, \ \forall t_i \in T_i, t_{-i} \in T_{-i}, \forall i \in N$$

Ex-post: Even after all agents have revealed their types, participating is weakly preferred.

In plain words

The auction is over. Every bid is public. Nobody regrets having walked into the room.

Implications of Individual Rationality

Lemma

In the single object allocation setting, consider a DSIC mechanism (f, p)

1. *It is IR iff* $\forall i \in N$ *and* $\forall t_{-i} \in T_{-i},\ p_i(0, t_{-i}) \leqslant 0$
2. *It is IR and satisfies no subsidy, i.e.,* $p_i(t_i, t_{-i}) \geqslant 0,\ \forall t_i \in T_i, t_{-i} \in T_{-i},\ \forall i \in N$ *iff* $\forall i \in N, t_{-i} \in T_{-i}, p_i(0, t_{-i}) = 0$

Implications of Individual Rationality

Lemma

In the single object allocation setting, consider a DSIC mechanism (f, p)

1. *It is IR iff* $\forall i \in N$ *and* $\forall t_{-i} \in T_{-i},\ p_i(0, t_{-i}) \leqslant 0$
2. *It is IR and satisfies no subsidy, i.e.,* $p_i(t_i, t_{-i}) \geqslant 0,\ \forall t_i \in T_i, t_{-i} \in T_{-i},\ \forall i \in N$ *iff* $\forall i \in N, t_{-i} \in T_{-i}, p_i(0, t_{-i}) = 0$

Proof

1. Suppose (f, p) is IR, then $0 - p_i(0, t_{-i}) \geqslant 0$, hence $p_i(0, t_{-i}) \leqslant 0$

Implications of Individual Rationality

Lemma

In the single object allocation setting, consider a DSIC mechanism (f, p)

1. *It is IR iff* $\forall i \in N$ *and* $\forall t_{-i} \in T_{-i},\ p_i(0, t_{-i}) \leqslant 0$
2. *It is IR and satisfies no subsidy, i.e.,* $p_i(t_i, t_{-i}) \geqslant 0,\ \forall t_i \in T_i, t_{-i} \in T_{-i},\ \forall i \in N$ *iff* $\forall i \in N, t_{-i} \in T_{-i}, p_i(0, t_{-i}) = 0$

Proof

1. Suppose (f, p) is IR, then $0 - p_i(0, t_{-i}) \geqslant 0$, hence $p_i(0, t_{-i}) \leqslant 0$
   Conversely, if $p_i(0, t_{-i}) \leqslant 0$, then the payoff of i is

$$t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) = t_i f_i(t_i, t_{-i}) - p_i(0, t_{-i}) - t_i f_i(t_i, t_{-i}) + \int_0^{t_i} f_i(x, t_{-i}) dx \geqslant 0$$

Implications of Individual Rationality

Lemma

In the single object allocation setting, consider a DSIC mechanism (f, p)

1. *It is IR iff* $\forall i \in N$ *and* $\forall t_{-i} \in T_{-i},\ p_i(0, t_{-i}) \leqslant 0$
2. *It is IR and satisfies no subsidy, i.e.,* $p_i(t_i, t_{-i}) \geqslant 0,\ \forall t_i \in T_i, t_{-i} \in T_{-i},\ \forall i \in N$ *iff* $\forall i \in N, t_{-i} \in T_{-i}, p_i(0, t_{-i}) = 0$

Proof

1. Suppose (f, p) is IR, then $0 - p_i(0, t_{-i}) \geqslant 0$, hence $p_i(0, t_{-i}) \leqslant 0$
   Conversely, if $p_i(0, t_{-i}) \leqslant 0$, then the payoff of i is

$$t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) = t_i f_i(t_i, t_{-i}) - p_i(0, t_{-i}) - t_i f_i(t_i, t_{-i}) + \int_0^{t_i} f_i(x, t_{-i}) dx \geqslant 0$$

2. IR $\implies p_i(0, t_{-i}) \leqslant 0$, if $p_i(t_i, t_{-i}) \geqslant 0\ \forall t_i \implies p_i(0, t_{-i}) \geqslant 0$; together, $p_i(0, t_{-i}) = 0$.

Implications of Individual Rationality

Lemma

In the single object allocation setting, consider a DSIC mechanism (f, p)

1. *It is IR iff* $\forall i \in N$ *and* $\forall t_{-i} \in T_{-i},\ p_i(0, t_{-i}) \leqslant 0$
2. *It is IR and satisfies no subsidy, i.e.,* $p_i(t_i, t_{-i}) \geqslant 0,\ \forall t_i \in T_i, t_{-i} \in T_{-i},\ \forall i \in N$ *iff* $\forall i \in N, t_{-i} \in T_{-i}, p_i(0, t_{-i}) = 0$

Proof

1. Suppose (f, p) is IR, then $0 - p_i(0, t_{-i}) \geqslant 0$, hence $p_i(0, t_{-i}) \leqslant 0$
   Conversely, if $p_i(0, t_{-i}) \leqslant 0$, then the payoff of i is

$$t_i f_i(t_i, t_{-i}) - p_i(t_i, t_{-i}) = t_i f_i(t_i, t_{-i}) - p_i(0, t_{-i}) - t_i f_i(t_i, t_{-i}) + \int_0^{t_i} f_i(x, t_{-i}) dx \geqslant 0$$

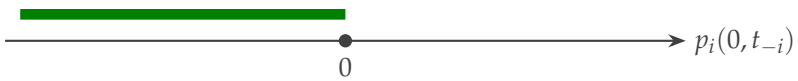
2. IR $\implies p_i(0, t_{-i}) \leqslant 0$, if $p_i(t_i, t_{-i}) \geqslant 0\ \forall t_i \implies p_i(0, t_{-i}) \geqslant 0$; together, $p_i(0, t_{-i}) = 0$.
   Clearly if $p_i(0, t_{-i}) = 0 \implies (f, p)$ is IR and no-subsidy.

The Lever $p_i(0, t_{-i})$



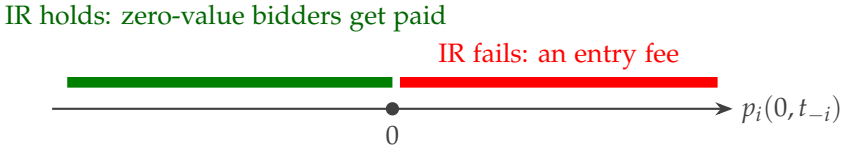
The Lever $p_i(0, t_{-i})$

IR holds: zero-value bidders get paid



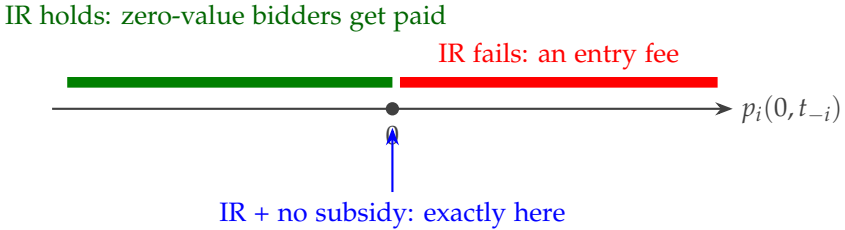
- A negative $p_i(0, t_{-i})$ is a gift: money flows **to** the bidder.

The Lever $p_i(0, t_{-i})$



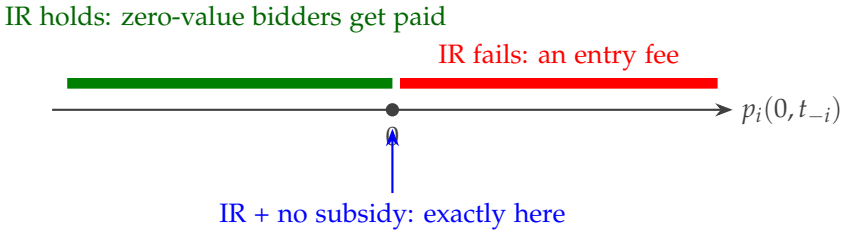
- A negative $p_i(0, t_{-i})$ is a gift: money flows **to** the bidder.

The Lever $p_i(0, t_{-i})$



- A negative $p_i(0, t_{-i})$ is a gift: money flows **to** the bidder.

The Lever $p_i(0, t_{-i})$



- A negative $p_i(0, t_{-i})$ is a gift: money flows **to** the bidder.
- The next two auctions use this gift to send money back to the bidders.

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

The object goes to the highest bidder, but the payment is such that everyone is compensated some amount. Assume, WLOG, $t_1 > t_2 > \ldots > t_n$

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

The object goes to the highest bidder, but the payment is such that everyone is compensated some amount. Assume, WLOG, $t_1 > t_2 > \ldots > t_n$

1. Highest and second highest bidders are compensated $\frac{1}{n}$ of the third highest bid.
   $p_1(0, t_{-1}) = p_2(0, t_{-2}) = -\frac{1}{n} t_3$

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

The object goes to the highest bidder, but the payment is such that everyone is compensated some amount. Assume, WLOG, $t_1 > t_2 > \ldots > t_n$

1. Highest and second highest bidders are compensated $\frac{1}{n}$ of the third highest bid.
   $p_1(0, t_{-1}) = p_2(0, t_{-2}) = -\frac{1}{n} t_3$
2. Everyone else receives $\frac{1}{n}$ of the second highest bid without him
   $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest in $\{t_j, j \neq i\}$

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

The object goes to the highest bidder, but the payment is such that everyone is compensated some amount. Assume, WLOG, $t_1 > t_2 > \ldots > t_n$

1. Highest and second highest bidders are compensated $\frac{1}{n}$ of the third highest bid.
   $p_1(0, t_{-1}) = p_2(0, t_{-2}) = -\frac{1}{n} t_3$
2. Everyone else receives $\frac{1}{n}$ of the second highest bid without him
   $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest in $\{t_j, j \neq i\}$

- 1 pays $= -\frac{1}{n} t_3 + t_1 - \int_0^{t_1} f_1(x, t_{-1}) dx = -\frac{1}{n} t_3 + t_1 - (t_1 - t_2) = -\frac{1}{n} t_3 + t_2$

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

The object goes to the highest bidder, but the payment is such that everyone is compensated some amount. Assume, WLOG, $t_1 > t_2 > \ldots > t_n$

1. Highest and second highest bidders are compensated $\frac{1}{n}$ of the third highest bid.
   $p_1(0, t_{-1}) = p_2(0, t_{-2}) = -\frac{1}{n} t_3$
2. Everyone else receives $\frac{1}{n}$ of the second highest bid without him
   $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest in $\{t_j, j \neq i\}$

- 1 pays $= -\frac{1}{n} t_3 + t_1 - \int_0^{t_1} f_1(x, t_{-1}) dx = -\frac{1}{n} t_3 + t_1 - (t_1 - t_2) = -\frac{1}{n} t_3 + t_2$
- 2 pays $= -\frac{1}{n} t_3$, all others $= -\frac{1}{n} t_2$

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

The object goes to the highest bidder, but the payment is such that everyone is compensated some amount. Assume, WLOG, $t_1 > t_2 > \ldots > t_n$

1. Highest and second highest bidders are compensated $\frac{1}{n}$ of the third highest bid.
   $p_1(0, t_{-1}) = p_2(0, t_{-2}) = -\frac{1}{n} t_3$
2. Everyone else receives $\frac{1}{n}$ of the second highest bid without him
   $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest in $\{t_j, j \neq i\}$

- 1 pays $= -\frac{1}{n} t_3 + t_1 - \int_0^{t_1} f_1(x, t_{-1}) dx = -\frac{1}{n} t_3 + t_1 - (t_1 - t_2) = -\frac{1}{n} t_3 + t_2$
- 2 pays $= -\frac{1}{n} t_3$, all others $= -\frac{1}{n} t_2$
- Total payment $= -\frac{1}{n} t_3 + t_2 - \frac{1}{n} t_3 - \frac{n-2}{n} t_2 = \frac{2}{n}(t_2 - t_3)$, which tends to 0 for large n.

Non-Vickrey Auctions: Example 1

Hackathon prize, redistributed: one GPU cluster, n teams. The organizer wants the money to stay with the teams.

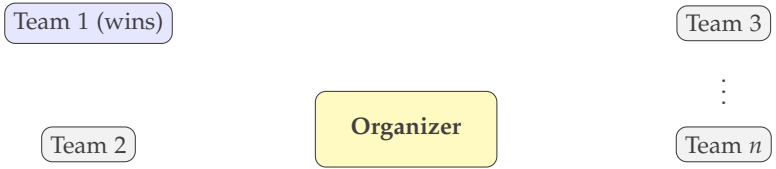
The object goes to the highest bidder, but the payment is such that everyone is compensated some amount. Assume, WLOG, $t_1 > t_2 > \ldots > t_n$

1. Highest and second highest bidders are compensated $\frac{1}{n}$ of the third highest bid.
   $p_1(0, t_{-1}) = p_2(0, t_{-2}) = -\frac{1}{n} t_3$
2. Everyone else receives $\frac{1}{n}$ of the second highest bid without him
   $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest in $\{t_j, j \neq i\}$

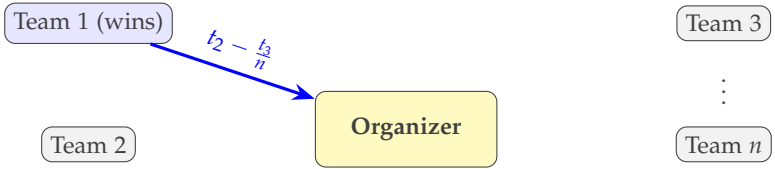
- 1 pays $= -\frac{1}{n} t_3 + t_1 - \int_0^{t_1} f_1(x, t_{-1}) dx = -\frac{1}{n} t_3 + t_1 - (t_1 - t_2) = -\frac{1}{n} t_3 + t_2$
- 2 pays $= -\frac{1}{n} t_3$, all others $= -\frac{1}{n} t_2$
- Total payment $= -\frac{1}{n} t_3 + t_2 - \frac{1}{n} t_3 - \frac{n-2}{n} t_2 = \frac{2}{n}(t_2 - t_3)$, which tends to 0 for large n.

Deterministic mechanism that redistributes the money

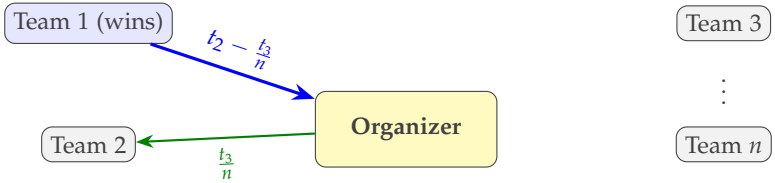
Where the Money Goes



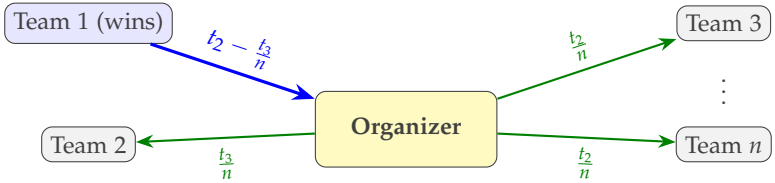
Where the Money Goes



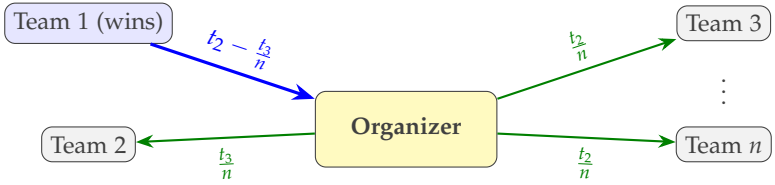
Where the Money Goes



Where the Money Goes

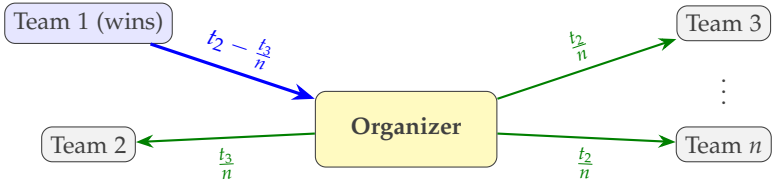


Where the Money Goes



Organizer keeps only $\frac{2}{n}(t_2 - t_3)$

Where the Money Goes



Why still DSIC?

Each team's rebate depends only on the *others'* bids. It is a $p_i(0, t_{-i})$ term. Myerson allows any such term.

Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.

Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.
- $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest bid in $\{t_j, j \neq i\}$

Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.
- $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest bid in $\{t_j, j \neq i\}$
- WLOG $t_1 > t_2 > \ldots > t_n$

Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.
- $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest bid in $\{t_j, j \neq i\}$
- WLOG $t_1 > t_2 > \ldots > t_n$
- 1 pays $= -\frac{1}{n}t_3 + (1 - \frac{1}{n})t_1 - \frac{1}{n}(t_2 - t_3) - (1 - \frac{1}{n})(t_1 - t_2) = (1 - \frac{2}{n})t_2$

Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.
- $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest bid in $\{t_j, j \neq i\}$
- WLOG $t_1 > t_2 > \ldots > t_n$
- 1 pays $= -\frac{1}{n}t_3 + (1 - \frac{1}{n})t_1 - \frac{1}{n}(t_2 - t_3) - (1 - \frac{1}{n})(t_1 - t_2) = (1 - \frac{2}{n})t_2$
- 2 pays $= -\frac{1}{n}t_3 + \frac{1}{n}t_2 - \frac{1}{n}(t_2 - t_3) = 0$

Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.
- $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest bid in $\{t_j, j \neq i\}$
- WLOG $t_1 > t_2 > \ldots > t_n$
- 1 pays $= -\frac{1}{n}t_3 + (1 - \frac{1}{n})t_1 - \frac{1}{n}(t_2 - t_3) - (1 - \frac{1}{n})(t_1 - t_2) = (1 - \frac{2}{n})t_2$
- 2 pays $= -\frac{1}{n}t_3 + \frac{1}{n}t_2 - \frac{1}{n}(t_2 - t_3) = 0$
- All others $= -\frac{1}{n}t_2$.

Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.
- $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest bid in $\{t_j, j \neq i\}$
- WLOG $t_1 > t_2 > \ldots > t_n$
- 1 pays $= -\frac{1}{n}t_3 + (1 - \frac{1}{n})t_1 - \frac{1}{n}(t_2 - t_3) - (1 - \frac{1}{n})(t_1 - t_2) = (1 - \frac{2}{n})t_2$
- 2 pays $= -\frac{1}{n}t_3 + \frac{1}{n}t_2 - \frac{1}{n}(t_2 - t_3) = 0$
- All others $= -\frac{1}{n}t_2$.
- Together = 0

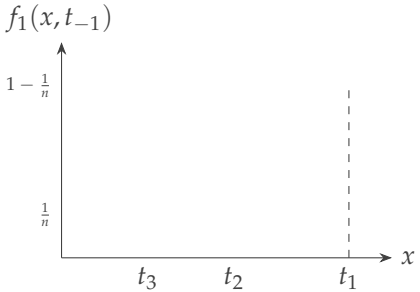
Non-Vickrey Auctions: Example 2

Hackathon prize, randomized: same teams, but the runner-up also gets a chance.

- Allocate the object w.p. $(1 - \frac{1}{n})$ to the highest bidder and w.p. $\frac{1}{n}$ to the second highest bidder.
- $p_i(0, t_{-i}) = -\frac{1}{n}$ second highest bid in $\{t_j, j \neq i\}$
- WLOG $t_1 > t_2 > \ldots > t_n$
- 1 pays $= -\frac{1}{n}t_3 + (1 - \frac{1}{n})t_1 - \frac{1}{n}(t_2 - t_3) - (1 - \frac{1}{n})(t_1 - t_2) = (1 - \frac{2}{n})t_2$
- 2 pays $= -\frac{1}{n}t_3 + \frac{1}{n}t_2 - \frac{1}{n}(t_2 - t_3) = 0$
- All others $= -\frac{1}{n}t_2$.
- Together = 0

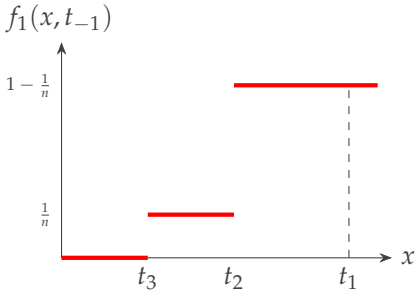
Randomized mechanism that redistributes the money

Team 1's Staircase



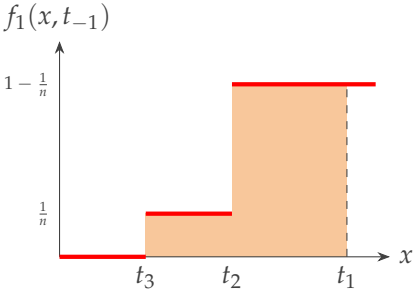
- Slide team 1's bid x up, others fixed.

Team 1's Staircase



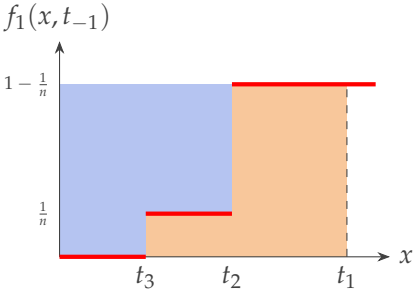
- Slide team 1's bid x up, others fixed.
- Below t_3: third or worse, no chance.
  Between t_3 and t_2: runner-up, chance $\frac{1}{n}$.
  Above t_2: winner, chance $1 - \frac{1}{n}$.

Team 1's Staircase



- Slide team 1's bid x up, others fixed.
- Below t_3: third or worse, no chance.
  Between t_3 and t_2: runner-up, chance $\frac{1}{n}$.
  Above t_2: winner, chance $1 - \frac{1}{n}$.
- Orange: the area under the staircase.

Team 1's Staircase



- Slide team 1's bid x up, others fixed.
- Below t_3: third or worse, no chance. Between t_3 and t_2: runner-up, chance $\frac{1}{n}$. Above t_2: winner, chance $1 - \frac{1}{n}$.
- Orange: the area under the staircase.
- Blue: rectangle minus orange. Add the rebate $-\frac{1}{n}t_3$. You get $(1 - \frac{2}{n})t_2$.

Chunk 4

Bayesian Mechanisms: BIC, NDE, and IIR

(≈ 20–25 minutes)

Revenue Maximization

Question

How to maximize revenue earned by the auctioneer?

Revenue Maximization

Question

How to maximize revenue earned by the auctioneer?

Question

Maximize w.r.t. what knowledge of the auctioneer?

Revenue Maximization

Question

How to maximize revenue earned by the auctioneer?

Question

Maximize w.r.t. what knowledge of the auctioneer?

Answer

The common prior distribution over types

Revenue Maximization

Question

How to maximize revenue earned by the auctioneer?

Question

Maximize w.r.t. what knowledge of the auctioneer?

Answer

The common prior distribution over types

Accordingly, the notions of incentive compatibility and individual rationality need to change

Revenue Maximization

Question

How to maximize revenue earned by the auctioneer?

Question

Maximize w.r.t. what knowledge of the auctioneer?

Answer

The common prior distribution over types

Accordingly, the notions of incentive compatibility and individual rationality need to change

The seller's view

A GPU startup has run hundreds of past auctions. It cannot see today's bids. But its logs tell it how bids are *distributed*. That distribution is the prior.

Bayesian setup



Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density

Bayesian setup



Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i

Bayesian setup



Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i
- Similarly, $g_{-i}(s_{-i}|s_i)$ is derived via Bayes rule from g

Bayesian setup

Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i
- Similarly, $g_{-i}(s_{-i}|s_i)$ is derived via Bayes rule from g
- Every mechanism $(f, p_1, p_2, \ldots, p_n)$ induces an expected allocation and payment rule (α, π)

$$\alpha_i(\underbrace{s_i}_{\text{reported}}; \underbrace{t_i}_{\text{true}}) =$$

where s_i is the reported type and t_i is the true type.

Bayesian setup

Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i
- Similarly, $g_{-i}(s_{-i}|s_i)$ is derived via Bayes rule from g
- Every mechanism $(f, p_1, p_2, \ldots, p_n)$ induces an expected allocation and payment rule (α, π)

$$\alpha_i(\underbrace{s_i}_{\text{reported}}; \underbrace{t_i}_{\text{true}}) = \underbrace{f_i(s_i, s_{-i})}_{\textbf{probabilistic allocation}}$$

where s_i is the reported type and t_i is the true type.

Bayesian setup

Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i
- Similarly, $g_{-i}(s_{-i}|s_i)$ is derived via Bayes rule from g
- Every mechanism $(f, p_1, p_2, \ldots, p_n)$ induces an expected allocation and payment rule (α, π)

$$\alpha_i(\underbrace{s_i}_{\text{reported}}; \underbrace{t_i}_{\text{true}}) = \int_{s_{-i} \in T_{-i}} \underbrace{f_i(s_i, s_{-i})}_{\text{probabilistic allocation}} \underbrace{g_{-i}(s_{-i}|t_i)}_{\text{prior on types of others}} ds_{-i}$$

where s_i is the reported type and t_i is the true type.

Bayesian setup

Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i
- Similarly, $g_{-i}(s_{-i}|s_i)$ is derived via Bayes rule from g
- Every mechanism $(f, p_1, p_2, \ldots, p_n)$ induces an expected allocation and payment rule (α, π)

$$\alpha_i(\underbrace{s_i}_{\text{reported}}; \underbrace{t_i}_{\text{true}}) = \int_{s_{-i} \in T_{-i}} \underbrace{f_i(s_i, s_{-i})}_{\text{probabilistic allocation}} \underbrace{g_{-i}(s_{-i}|t_i)}_{\text{prior on types of others}} ds_{-i} \quad \textbf{expected allocation}$$

where s_i is the reported type and t_i is the true type.

Bayesian setup

Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i
- Similarly, $g_{-i}(s_{-i}|s_i)$ is derived via Bayes rule from g
- Every mechanism $(f, p_1, p_2, \ldots, p_n)$ induces an expected allocation and payment rule (α, π)

$$\alpha_i(\underbrace{s_i}_{\text{reported}}; \underbrace{t_i}_{\text{true}}) = \int_{s_{-i} \in T_{-i}} \underbrace{f_i(s_i, s_{-i})}_{\text{probabilistic allocation}} \underbrace{g_{-i}(s_{-i}|t_i)}_{\text{prior on types of others}} ds_{-i} \quad \textbf{expected allocation}$$

$$\pi_i(s_i; t_i) = \int_{s_{-i} \in T_{-i}} p_i(s_i, s_{-i}) g_{-i}(s_{-i}|t_i) ds_{-i} \quad \textbf{expected payment}$$

where s_i is the reported type and t_i is the true type.

Bayesian setup

Preliminaries

- $T_i = [0, b_i]$, Common prior G over $T = \times_{i=1}^{n} T_i$, g denotes the density
- $G_{-i}(s_{-i}|s_i)$ is the conditional distribution over s_{-i}, given i's type is s_i
- Similarly, $g_{-i}(s_{-i}|s_i)$ is derived via Bayes rule from g
- Every mechanism $(f, p_1, p_2, \ldots, p_n)$ induces an expected allocation and payment rule (α, π)

$$\alpha_i(\underbrace{s_i}_{\text{reported}}; \underbrace{t_i}_{\text{true}}) = \int_{s_{-i} \in T_{-i}} \underbrace{f_i(s_i, s_{-i})}_{\text{probabilistic allocation}} \underbrace{g_{-i}(s_{-i}|t_i)}_{\text{prior on types of others}} ds_{-i} \quad \textbf{expected allocation}$$

$$\pi_i(s_i; t_i) = \int_{s_{-i} \in T_{-i}} p_i(s_i, s_{-i}) g_{-i}(s_{-i}|t_i) ds_{-i} \quad \textbf{expected payment}$$

where s_i is the reported type and t_i is the true type.

- Expected utility of agent i: $u_i(s_i; t_i) = t_i \alpha_i(s_i; t_i) - \pi_i(s_i; t_i)$

Bayesian Incentive Compatibility

Definition (Bayesian Incentive Compatibility (BIC))

A mechanism (f, p) is Bayesian incentive compatible (BIC) if $\forall i \in N, \forall s_i, t_i \in T_i$

$$t_i \alpha_i(t_i; t_i) - \pi_i(t_i; t_i) \geqslant t_i \alpha_i(s_i; t_i) - \pi_i(s_i; t_i)$$

Bayesian Incentive Compatibility

Definition (Bayesian Incentive Compatibility (BIC))

A mechanism (f, p) is Bayesian incentive compatible (BIC) if $\forall i \in N, \forall s_i, t_i \in T_i$

$$t_i \alpha_i(t_i; t_i) - \pi_i(t_i; t_i) \geqslant t_i \alpha_i(s_i; t_i) - \pi_i(s_i; t_i)$$

Similarly, f is Bayesian implementable if $\exists p$ s.t. (f, p) is BIC.

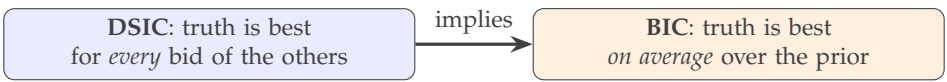
Bayesian Incentive Compatibility

Definition (Bayesian Incentive Compatibility (BIC))

A mechanism (f, p) is Bayesian incentive compatible (BIC) if $\forall i \in N, \forall s_i, t_i \in T_i$

$$t_i \alpha_i(t_i; t_i) - \pi_i(t_i; t_i) \geqslant t_i \alpha_i(s_i; t_i) - \pi_i(s_i; t_i)$$

Similarly, f is Bayesian implementable if $\exists p$ s.t. (f, p) is BIC.



Characterization of BIC mechanisms

Assume that **priors are independent**, i.e., agent i's value is drawn from a distribution G_i (density g_i) independently from other agents.

$$G(s_1, s_2, \ldots, s_n) = \prod_{i \in N} G_i(s_i), \qquad G(s_{-i}|t_i) = \prod_{j \neq i} G_j(s_j)$$

Characterization of BIC mechanisms

Assume that **priors are independent**, i.e., agent i's value is drawn from a distribution G_i (density g_i) independently from other agents.

$$G(s_1, s_2, \ldots, s_n) = \prod_{i \in N} G_i(s_i), \qquad G(s_{-i}|t_i) = \prod_{j \neq i} G_j(s_j)$$

Also we will use the shorthand notation assuming the agents are revealing true types, unless otherwise stated, i.e., $\alpha(t_i) = \alpha(t_i; t_i)$

Characterization of BIC mechanisms

Assume that **priors are independent**, i.e., agent i's value is drawn from a distribution G_i (density g_i) independently from other agents.

$$G(s_1, s_2, \ldots, s_n) = \prod_{i \in N} G_i(s_i), \qquad G(s_{-i}|t_i) = \prod_{j \neq i} G_j(s_j)$$

Also we will use the shorthand notation assuming the agents are revealing true types, unless otherwise stated, i.e., $\alpha(t_i) = \alpha(t_i; t_i)$

Definition

An allocation rule is **Non-decreasing in expectation (NDE)** if $\forall i \in N, \forall s_i, t_i \in T_i$ with $s_i < t_i$ we have $\alpha_i(s_i) \leqslant \alpha_i(t_i)$

Characterization of BIC mechanisms

Assume that **priors are independent**, i.e., agent i's value is drawn from a distribution G_i (density g_i) independently from other agents.

$$G(s_1, s_2, \ldots, s_n) = \prod_{i \in N} G_i(s_i), \qquad G(s_{-i}|t_i) = \prod_{j \neq i} G_j(s_j)$$

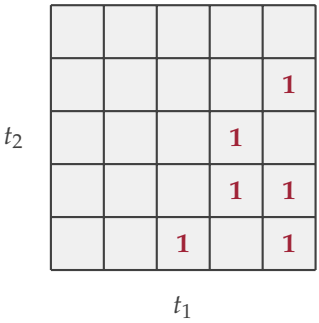
Also we will use the shorthand notation assuming the agents are revealing true types, unless otherwise stated, i.e., $\alpha(t_i) = \alpha(t_i; t_i)$

Definition

An allocation rule is **Non-decreasing in expectation (NDE)** if $\forall i \in N, \forall s_i, t_i \in T_i$ with $s_i < t_i$ we have $\alpha_i(s_i) \leqslant \alpha_i(t_i)$

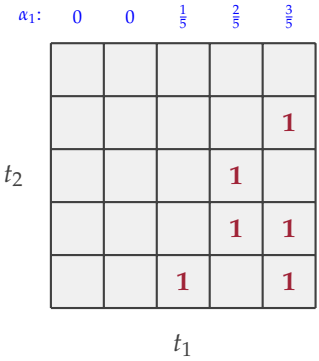
Note: The rules that are non-decreasing (defined before) are always NDE, but there can be more rules that are NDE

NDE but not ND



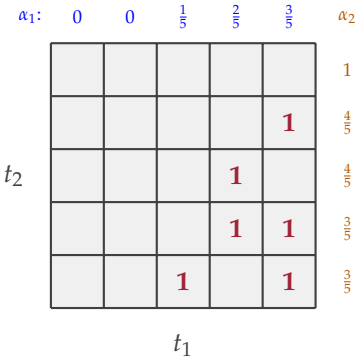
- All five types are equally likely, for both agents. A **1** means agent 1 wins. Otherwise agent 2 wins.

NDE but not ND



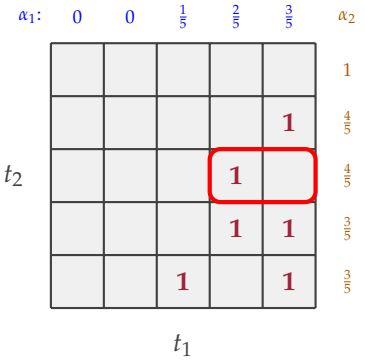
- All five types are equally likely, for both agents. A **1** means agent 1 wins. Otherwise agent 2 wins.
- Average each column: $\alpha_1(t_1)$ never goes down.

NDE but not ND



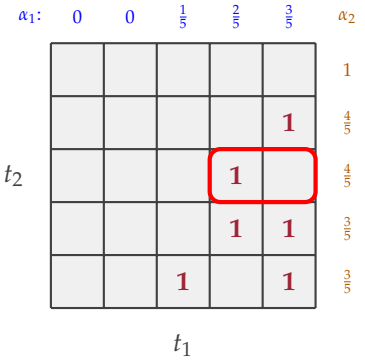
- All five types are equally likely, for both agents. A **1** means agent 1 wins. Otherwise agent 2 wins.
- Average each column: $\alpha_1(t_1)$ never goes down.
- Average each row: $\alpha_2(t_2)$ never goes down.

NDE but not ND



- All five types are equally likely, for both agents. A **1** means agent 1 wins. Otherwise agent 2 wins.
- Average each column: $\alpha_1(t_1)$ never goes down.
- Average each row: $\alpha_2(t_2)$ never goes down.
- But look at row 3. Agent 1 wins at column 4, and **loses** at column 5.

NDE but not ND



- All five types are equally likely, for both agents. A **1** means agent 1 wins. Otherwise agent 2 wins.
- Average each column: $\alpha_1(t_1)$ never goes down.
- Average each row: $\alpha_2(t_2)$ never goes down.
- But look at row 3. Agent 1 wins at column 4, and **loses** at column 5.
- So $\alpha_1(t_1)$ and $\alpha_2(t_2)$ are monotone, but $f(t_1, t_2)$ is not.

Bayesian Myerson

Theorem (Myerson 1981)

A mechanism (f, p) in the independent prior setting is BIC iff

- *f is NDE, and*
- *p_i satisfies $\pi_i(t_i) = \pi_i(0) + t_i\alpha_i(t_i) - \int_0^{t_i} \alpha_i(x)\,dx, \quad \forall t_i \in T_i, \forall i \in N$*

Bayesian Myerson

Theorem (Myerson 1981)

A mechanism (f, p) in the independent prior setting is BIC iff

- *f is NDE, and*
- *p_i satisfies $\pi_i(t_i) = \pi_i(0) + t_i\alpha_i(t_i) - \int_0^{t_i} \alpha_i(x)\,dx, \quad \forall t_i \in T_i, \forall i \in N$*

Proof.

This is Bayesian version of the earlier Myerson theorem, proof proceeds in similar lines as before [exercise] □

Bayesian Myerson

Theorem (Myerson 1981)

A mechanism (f, p) in the independent prior setting is BIC iff

- *f is NDE, and*
- *p_i satisfies $\pi_i(t_i) = \pi_i(0) + t_i \alpha_i(t_i) - \int_0^{t_i} \alpha_i(x)\, dx, \quad \forall t_i \in T_i, \forall i \in N$*

Proof.

This is Bayesian version of the earlier Myerson theorem, proof proceeds in similar lines as before [exercise] □

Same picture, new curve

Replace the blue curve $f_i(\cdot, t_{-i})$ by the expected curve $\alpha_i(\cdot)$. The pay and keep areas work exactly as before.

Characterization of BIC rules

As we are in the Bayesian setting now, we can define an analog of individual rationality.

Definition

A mechanism (f, p) is **interim individually rational (IIR)** if for every bidder $i \in N$, we have $t_i \alpha_i(t_i) - \pi_i(t_i) \geqslant 0$, $\forall t_i \in T_i$

Characterization of BIC rules

As we are in the Bayesian setting now, we can define an analog of individual rationality.

Definition

A mechanism (f, p) is **interim individually rational (IIR)** if for every bidder $i \in N$, we have $t_i \alpha_i(t_i) - \pi_i(t_i) \geqslant 0$, $\forall t_i \in T_i$

Interim: you know your own type, but not the others'. On average, joining is weakly better.

Characterization of BIC rules

As we are in the Bayesian setting now, we can define an analog of individual rationality.

Definition

A mechanism (f, p) is **interim individually rational (IIR)** if for every bidder $i \in N$, we have $t_i \alpha_i(t_i) - \pi_i(t_i) \geqslant 0,\ \forall t_i \in T_i$

Interim: you know your own type, but not the others'. On average, joining is weakly better.

Lemma

A mechanism (f, p) is BIC and IIR iff

- *f is NDE*
- *p_i satisfies $\pi_i(t_i) = \pi_i(0) + t_i \alpha_i(t_i) - \int_0^{t_i} \alpha_i(x)\, dx,\ \ \forall t_i \in T_i, \forall i \in N$*
- *$\forall i \in N,\ \pi_i(0) \leqslant 0$*

Characterization of BIC rules

Proof-sketch:

- The first two conditions uniquely identify a BIC mechanism

Characterization of BIC rules

Proof-sketch:

- The first two conditions uniquely identify a BIC mechanism
- So, the proof requires to show that IIR along with first two conditions is equivalent to third condition

Characterization of BIC rules

Proof-sketch:

- The first two conditions uniquely identify a BIC mechanism
- So, the proof requires to show that IIR along with first two conditions is equivalent to third condition
- **Forward direction:** apply IIR at $t_i = 0$ on second condition and get $\pi_i(0) \leqslant 0$

Characterization of BIC rules

Proof-sketch:

- The first two conditions uniquely identify a BIC mechanism
- So, the proof requires to show that IIR along with first two conditions is equivalent to third condition
- **Forward direction:** apply IIR at $t_i = 0$ on second condition and get $\pi_i(0) \leqslant 0$
- **Reverse direction:** $t_i \alpha_i(t_i) - \pi_i(t_i) = -\pi_i(0) + \int_0^{t_i} \alpha_i(s_i)\, ds_i \geqslant 0$ if $\pi_i(0) \leqslant 0$

Characterization of BIC rules

Proof-sketch:

- The first two conditions uniquely identify a BIC mechanism
- So, the proof requires to show that IIR along with first two conditions is equivalent to third condition
- **Forward direction:** apply IIR at $t_i = 0$ on second condition and get $\pi_i(0) \leqslant 0$
- **Reverse direction:** $t_i \alpha_i(t_i) - \pi_i(t_i) = -\pi_i(0) + \int_0^{t_i} \alpha_i(s_i)\, ds_i \geqslant 0$ if $\pi_i(0) \leqslant 0$

The mining-pool entry fee

A block-building auction charges every bidder a fee upfront, even one who values the slot at 0. That is $\pi_i(0) > 0$. The zero-value bidder walks away with a negative expected utility. IIR fails.



In-Class Quiz 2

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.

Lecture 22 and Week 11: What We Built

- Myerson's lemma: DSIC $\iff$ non-decreasing allocation + the area payment formula.

Lecture 22 and Week 11: What We Built

- Myerson's lemma: DSIC $\iff$ non-decreasing allocation + the area payment formula.
- Forward: DSIC makes utility convex, with the win probability as its subgradient.

Lecture 22 and Week 11: What We Built

- Myerson's lemma: DSIC $\iff$ non-decreasing allocation + the area payment formula.
- Forward: DSIC makes utility convex, with the win probability as its subgradient.
- Reverse: a monotone curve makes every lie cost a non-negative area.

Lecture 22 and Week 11: What We Built

- Myerson's lemma: DSIC $\iff$ non-decreasing allocation + the area payment formula.
- Forward: DSIC makes utility convex, with the win probability as its subgradient.
- Reverse: a monotone curve makes every lie cost a non-negative area.
- Vickrey, reserve prices and the odd seller all fit. Payment = the threshold.

Lecture 22 and Week 11: What We Built



- Myerson's lemma: DSIC $\iff$ non-decreasing allocation + the area payment formula.
- Forward: DSIC makes utility convex, with the win probability as its subgradient.
- Reverse: a monotone curve makes every lie cost a non-negative area.
- Vickrey, reserve prices and the odd seller all fit. Payment = the threshold.
- IR $\iff$ $p_i(0, t_{-i}) \leqslant 0$. Redistribution auctions spend that slack to return money.

Lecture 22 and Week 11: What We Built

- Myerson's lemma: DSIC $\iff$ non-decreasing allocation + the area payment formula.
- Forward: DSIC makes utility convex, with the win probability as its subgradient.
- Reverse: a monotone curve makes every lie cost a non-negative area.
- Vickrey, reserve prices and the odd seller all fit. Payment = the threshold.
- IR $\iff$ $p_i(0, t_{-i}) \leqslant 0$. Redistribution auctions spend that slack to return money.
- Bayesian: BIC $\iff$ NDE + expected payment formula. IIR $\iff$ additionally $\pi_i(0) \leqslant 0$.

Lecture 22 and Week 11: What We Built



- Myerson's lemma: DSIC $\iff$ non-decreasing allocation + the area payment formula.
- Forward: DSIC makes utility convex, with the win probability as its subgradient.
- Reverse: a monotone curve makes every lie cost a non-negative area.
- Vickrey, reserve prices and the odd seller all fit. Payment = the threshold.
- IR $\iff$ $p_i(0, t_{-i}) \leqslant 0$. Redistribution auctions spend that slack to return money.
- Bayesian: BIC $\iff$ NDE + expected payment formula. IIR $\iff$ additionally $\pi_i(0) \leqslant 0$.
- **Next week:** the revenue-optimal auction.



End of Lecture 22

See you next class.



भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay