

भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay

CS 6001: Game Theory and Algorithmic Mechanism Design

Week 9 – The Median Voter Theorem, Task Sharing, and Mechanisms with Money

Swaprava Nath

October, 2026

ज्ञानम् परमम् ध्येयम्

Knowledge is the supreme goal

Week 9 Roadmap

Lecture	Chunks (20–25 min each)	Quizzes
Lecture 17	1. Building the Phantom Peaks 2. Case 1: The Median Is a Phantom 3. Case 2: The Median Is a Team's Peak 4. Sharing a Divisible Task	Quiz 1 after Chunk 2 Quiz 2 after Chunk 4
Lecture 18	1. Fair and Truthful Splits 2. The Uniform Rule 3. Money Enters: Quasi-Linear Preferences 4. DSIC, Efficiency, and Groves Payments	Quiz 3 after Chunk 2 Quiz 4 after Chunk 4

Contents



- Lecture 17: The Median Voter Theorem and Task Sharing
- Lecture 18: The Uniform Rule and Mechanisms with Money



Chunk 1

Building the Phantom Peaks

(≈ 20 minutes)

Where We Stopped

Three teams share one GPU cluster. They must agree on one **power cap**.

Where We Stopped

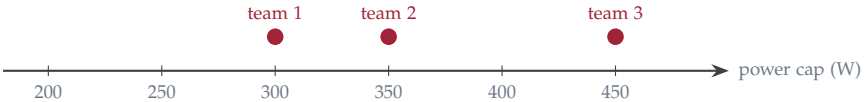
Three teams share one GPU cluster. They must agree on one **power cap**.

Each team has an ideal cap. Farther away is worse, on both sides. So preferences are **single-peaked**.

Where We Stopped

Three teams share one GPU cluster. They must agree on one **power cap**.

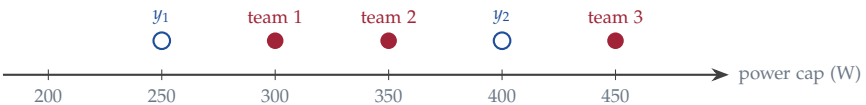
Each team has an ideal cap. Farther away is worse, on both sides. So preferences are **single-peaked**.



Where We Stopped

Three teams share one GPU cluster. They must agree on one **power cap**.

Each team has an ideal cap. Farther away is worse, on both sides. So preferences are **single-peaked**.

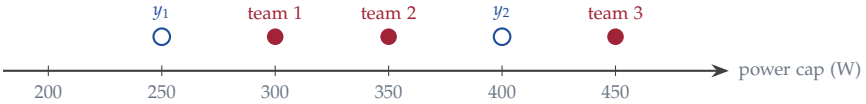


A **median voter SCF** adds $n-1$ fixed **phantom peaks** B, and picks the median of all $2n-1$ points.

Where We Stopped

Three teams share one GPU cluster. They must agree on one **power cap**.

Each team has an ideal cap. Farther away is worse, on both sides. So preferences are **single-peaked**.



A **median voter SCF** adds $n-1$ fixed **phantom peaks** B, and picks the median of all $2n-1$ points.

Theorem (Moulin, 1980)

A strategyproof SCF f is ONTO and anonymous if and only if it is a median voter SCF.

The Easy Direction ($\Longleftarrow$)

Let $f(P) = \text{median}(B, \text{peaks}(P))$. Why is it SP, ANON, and ONTO?

- **SP:** we proved it last week. To move the median, a team must report a peak on the *other* side of it. That only pushes the outcome away from its true peak.

The Easy Direction ($\Longleftarrow$)

Let $f(P) = \text{median}(B, \text{peaks}(P))$. Why is it SP, ANON, and ONTO?

- **SP:** we proved it last week. To move the median, a team must report a peak on the *other* side of it. That only pushes the outcome away from its true peak.
- **ANON:** rename the teams. The set of peaks does not change. So the median does not change.

The Easy Direction ($\Longleftarrow$)

Let $f(P) = \text{median}(B, \text{peaks}(P))$. Why is it SP, ANON, and ONTO?

- **SP:** we proved it last week. To move the median, a team must report a peak on the *other* side of it. That only pushes the outcome away from its true peak.
- **ANON:** rename the teams. The set of peaks does not change. So the median does not change.
- **ONTO:** pick any cap a. Let every team report peak a.

The Easy Direction ($\Longleftarrow$)

Let $f(P) = \text{median}(B, \text{peaks}(P))$. Why is it SP, ANON, and ONTO?

- **SP:** we proved it last week. To move the median, a team must report a peak on the *other* side of it. That only pushes the outcome away from its true peak.
- **ANON:** rename the teams. The set of peaks does not change. So the median does not change.
- **ONTO:** pick any cap a. Let every team report peak a.
  - n of the $2n - 1$ points now sit at a.

The Easy Direction ($\Longleftarrow$)

Let $f(P) = \text{median}(B, \text{peaks}(P))$. Why is it SP, ANON, and ONTO?

- **SP:** we proved it last week. To move the median, a team must report a peak on the *other* side of it. That only pushes the outcome away from its true peak.
- **ANON:** rename the teams. The set of peaks does not change. So the median does not change.
- **ONTO:** pick any cap a. Let every team report peak a.
  - n of the $2n - 1$ points now sit at a.
  - That is a majority. The median is a, wherever the phantoms are.

The Easy Direction ($\Longleftarrow$)

Let $f(P) = \text{median}(B, \text{peaks}(P))$. Why is it SP, ANON, and ONTO?

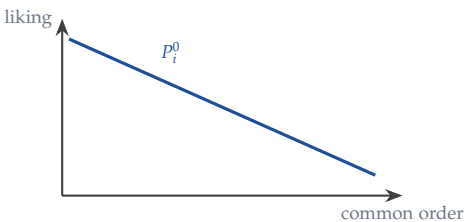
- **SP:** we proved it last week. To move the median, a team must report a peak on the *other* side of it. That only pushes the outcome away from its true peak.
- **ANON:** rename the teams. The set of peaks does not change. So the median does not change.
- **ONTO:** pick any cap a. Let every team report peak a.
  - n of the $2n - 1$ points now sit at a.
  - That is a majority. The median is a, wherever the phantoms are.

The real work

The other direction. Given *any* f that is SP, ANON, ONTO, we must *find* phantoms that make f a median rule.

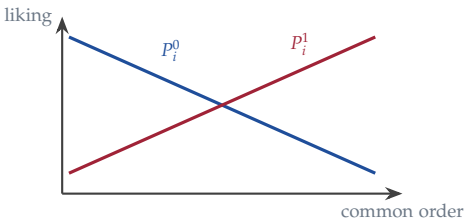
Two Extreme Reports

- P_i^0: team i's peak is the **leftmost** alternative. It wants the lowest cap.



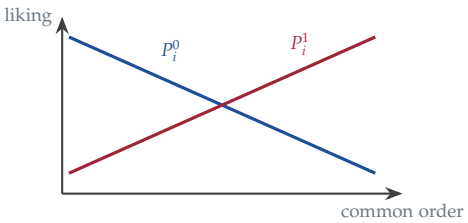
Two Extreme Reports

- P_i^0: team i's peak is the **leftmost** alternative. It wants the lowest cap.
- P_i^1: team i's peak is the **rightmost** alternative. It wants the highest cap.



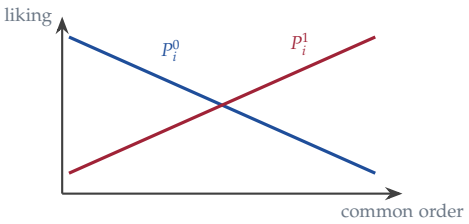
Two Extreme Reports

- P_i^0: team i's peak is the **leftmost** alternative. It wants the lowest cap.
- P_i^1: team i's peak is the **rightmost** alternative. It wants the highest cap.
- Both are single-peaked. So both are valid reports.



Two Extreme Reports

- P_i^0: team i's peak is the **leftmost** alternative. It wants the lowest cap.
- P_i^1: team i's peak is the **rightmost** alternative. It wants the highest cap.
- Both are single-peaked. So both are valid reports.
- Under P_i^0, smaller is always better. Under P_i^1, larger is always better.



Constructing the Phantom Peaks

Use f itself. Send $n-j$ teams to the far left and j teams to the far right:

$$y_j = f\big(\underbrace{P_1^0, \ldots, P_{n-j}^0}_{n-j \text{ leftmost}}, \underbrace{P_{n-j+1}^1, \ldots, P_n^1}_{j \text{ rightmost}}\big), \quad j = 1, \ldots, n-1.$$

Constructing the Phantom Peaks

Use f itself. Send $n-j$ teams to the far left and j teams to the far right:

$$y_j = f\big(\underbrace{P^0_1, \ldots, P^0_{n-j}}_{n-j \text{ leftmost}}, \underbrace{P^1_{n-j+1}, \ldots, P^1_n}_{j \text{ rightmost}}\big), \quad j = 1, \ldots, n-1.$$



$n = 4$ teams: one more team jumps to the right at every step

Constructing the Phantom Peaks

Use f itself. Send $n-j$ teams to the far left and j teams to the far right:

$$y_j = f\big(\underbrace{P^0_1, \ldots, P^0_{n-j}}_{n-j \text{ leftmost}}, \underbrace{P^1_{n-j+1}, \ldots, P^1_n}_{j \text{ rightmost}}\big), \quad j = 1, \ldots, n-1.$$



$n = 4$ teams: one more team jumps to the right at every step

By **anonymity**, only the *count* j matters, not which teams move.

The Phantoms Come Out Sorted



Claim

$y_j \leqslant y_{j+1}$ *for every* $j = 1, \ldots, n-2$.

The Phantoms Come Out Sorted

Claim

$y_j \leqslant y_{j+1}$ *for every* $j = 1, \ldots, n-2$.

Proof. Put the two profiles side by side:

$$y_j = f(P_1^0, \ldots, P_{n-j-1}^0, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1)$$
$$y_{j+1} = f(P_1^0, \ldots, P_{n-j-1}^0, P_{n-j}^1, P_{n-j+1}^1, \ldots, P_n^1)$$

- Only team $n-j$ changed its report.

The Phantoms Come Out Sorted

Claim

$y_j \leqslant y_{j+1}$ *for every* $j = 1, \ldots, n-2$.

Proof. Put the two profiles side by side:

$$y_j = f(P^0_1, \ldots, P^0_{n-j-1}, P^0_{n-j}, P^1_{n-j+1}, \ldots, P^1_n)$$
$$y_{j+1} = f(P^0_1, \ldots, P^0_{n-j-1}, P^1_{n-j}, P^1_{n-j+1}, \ldots, P^1_n)$$

- Only team $n-j$ changed its report.
- Say its true preference is P^0_{n-j}. By SP, lying to P^1_{n-j} does not help: $y_j \; R^0_{n-j} \; y_{j+1}$.

The Phantoms Come Out Sorted

Claim

$y_j \leqslant y_{j+1}$ *for every* $j = 1, \ldots, n-2$.

Proof. Put the two profiles side by side:

$$y_j = f(P^0_1, \ldots, P^0_{n-j-1}, P^0_{n-j}, P^1_{n-j+1}, \ldots, P^1_n)$$
$$y_{j+1} = f(P^0_1, \ldots, P^0_{n-j-1}, P^1_{n-j}, P^1_{n-j+1}, \ldots, P^1_n)$$

- Only team $n-j$ changed its report.
- Say its true preference is P^0_{n-j}. By SP, lying to P^1_{n-j} does not help: $y_j \; R^0_{n-j} \; y_{j+1}$.
- P^0_{n-j} likes smaller values more. So $y_j \leqslant y_{j+1}$.

□

The Phantoms Come Out Sorted

Claim

$y_j \leqslant y_{j+1}$ *for every* $j = 1, \ldots, n-2$.

Proof. Put the two profiles side by side:

$$y_j = f(P^0_1, \ldots, P^0_{n-j-1}, P^0_{n-j}, P^1_{n-j+1}, \ldots, P^1_n)$$
$$y_{j+1} = f(P^0_1, \ldots, P^0_{n-j-1}, P^1_{n-j}, P^1_{n-j+1}, \ldots, P^1_n)$$

- Only team $n-j$ changed its report.
- Say its true preference is P^0_{n-j}. By SP, lying to P^1_{n-j} does not help: $y_j \; R^0_{n-j} \; y_{j+1}$.
- P^0_{n-j} likes smaller values more. So $y_j \leqslant y_{j+1}$. □

Now define the candidate rule: $g(P) = \text{median}(y_1, \ldots, y_{n-1}, p_1, \ldots, p_n)$. Goal: $f \equiv g$.



Chunk 2

Case 1: The Median Is a Phantom

(≈ 20 minutes)

Setting Up the Two Cases

Fix any profile P with peaks $p_1, \ldots, p_n$. By ANON, relabel so that $p_1 \leqslant \cdots \leqslant p_n$.

Setting Up the Two Cases

Fix any profile P with peaks $p_1, \ldots, p_n$. By ANON, relabel so that $p_1 \leqslant \cdots \leqslant p_n$.

Let $a = \text{median}(y_1, \ldots, y_{n-1}, p_1, \ldots, p_n)$. Two cases:

- **Case 1:** a is a phantom peak, $a = y_j$
- **Case 2:** a is a team's peak

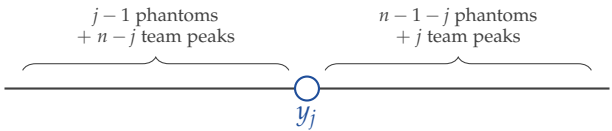
Setting Up the Two Cases

Fix any profile P with peaks $p_1, \ldots, p_n$. By ANON, relabel so that $p_1 \leqslant \cdots \leqslant p_n$.

Let $a = \text{median}(y_1, \ldots, y_{n-1}, p_1, \ldots, p_n)$. Two cases:

- **Case 1:** a is a phantom peak, $a = y_j$
- **Case 2:** a is a team's peak

Case 1, counting. The median has $n - 1$ points on each side.



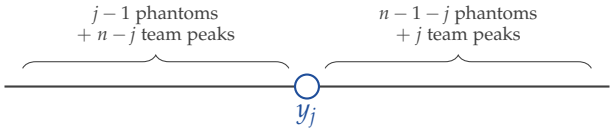
Setting Up the Two Cases

Fix any profile P with peaks $p_1, \ldots, p_n$. By ANON, relabel so that $p_1 \leqslant \cdots \leqslant p_n$.

Let $a = \text{median}(y_1, \ldots, y_{n-1}, p_1, \ldots, p_n)$. Two cases:

- **Case 1:** a is a phantom peak, $a = y_j$
- **Case 2:** a is a team's peak

Case 1, counting. The median has $n - 1$ points on each side.



So $p_1 \leqslant \cdots \leqslant p_{n-j} \leqslant y_j \leqslant p_{n-j+1} \leqslant \cdots \leqslant p_n$.

Case 1: Walk the Left Teams In

Start from the profile that *defines* y_j. Swap in one true preference at a time.

Case 1: Walk the Left Teams In

Start from the profile that *defines* y_j. Swap in one true preference at a time.

$$f(P_1^0, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = y_j \quad \text{(definition)}$$

Case 1: Walk the Left Teams In

Start from the profile that *defines* y_j. Swap in one true preference at a time.

$$f(P_1^0, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = y_j \quad \text{(definition)}$$
$$f(P_1, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = b \quad \text{(say)}$$

Case 1: Walk the Left Teams In

Start from the profile that *defines* y_j. Swap in one true preference at a time.

$$f(P_1^0, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = y_j \quad \text{(definition)}$$
$$f(P_1, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = b \quad \text{(say)}$$

- SP, true type P_1^0: $y_j \; R_1^0 \; b$. Leftmost peak, so $y_j \leqslant b$.

Case 1: Walk the Left Teams In

Start from the profile that *defines* y_j. Swap in one true preference at a time.

$$f(P_1^0, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = y_j \quad \text{(definition)}$$
$$f(P_1, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = b \quad \text{(say)}$$

- SP, true type P_1^0: $y_j \; R_1^0 \; b$. Leftmost peak, so $y_j \leqslant b$.
- SP, true type P_1: $b \; R_1 \; y_j$. But $p_1 \leqslant y_j \leqslant b$, all on one side. Single-peaked, so $b \leqslant y_j$.

Case 1: Walk the Left Teams In

Start from the profile that *defines* y_j. Swap in one true preference at a time.

$$f(P_1^0, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = y_j \quad \text{(definition)}$$
$$f(P_1, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = b \quad \text{(say)}$$

- SP, true type P_1^0: $y_j \; R_1^0 \; b$. Leftmost peak, so $y_j \leqslant b$.
- SP, true type P_1: $b \; R_1 \; y_j$. But $p_1 \leqslant y_j \leqslant b$, all on one side. Single-peaked, so $b \leqslant y_j$.
- Hence $b = y_j$.

Case 1: Walk the Left Teams In

Start from the profile that *defines* y_j. Swap in one true preference at a time.

$$f(P_1^0, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = y_j \quad \text{(definition)}$$
$$f(P_1, P_2^0, \ldots, P_{n-j}^0, P_{n-j+1}^1, \ldots, P_n^1) = b \quad \text{(say)}$$

- SP, true type P_1^0: $y_j \; R_1^0 \; b$. Leftmost peak, so $y_j \leqslant b$.
- SP, true type P_1: $b \; R_1 \; y_j$. But $p_1 \leqslant y_j \leqslant b$, all on one side. Single-peaked, so $b \leqslant y_j$.
- Hence $b = y_j$.

Repeat for teams $2, \ldots, n-j$. Each has $p_k \leqslant y_j$, so the same two lines work:

$$f(P_1, \ldots, P_{n-j}, P_{n-j+1}^1, \ldots, P_n^1) = y_j.$$

Case 1: Walk the Right Teams In

Now swap team n from P_n^1 to its true P_n:

$$f(P_1, \ldots, P_{n-j}, P_{n-j+1}^1, \ldots, P_n) = b \quad \text{(say)}$$

Case 1: Walk the Right Teams In

Now swap team n from P_n^1 to its true P_n:

$$f(P_1, \ldots, P_{n-j}, P_{n-j+1}^1, \ldots, P_n) = b \quad \text{(say)}$$

$$\left.\begin{array}{l} y_j \; R_n^1 \; b \text{ and } P_n^1 \text{ peaks rightmost} \implies b \leqslant y_j \\ b \; R_n \; y_j \text{ and } b \leqslant y_j \leqslant p_n \implies y_j \leqslant b \end{array}\right\} \; b = y_j$$

Case 1: Walk the Right Teams In

Now swap team n from P_n^1 to its true P_n:

$$f(P_1, \ldots, P_{n-j}, P_{n-j+1}^1, \ldots, P_n) = b \quad \text{(say)}$$

$$\left.\begin{array}{l} y_j \; R_n^1 \; b \text{ and } P_n^1 \text{ peaks rightmost} \implies b \leqslant y_j \\ b \; R_n \; y_j \text{ and } b \leqslant y_j \leqslant p_n \implies y_j \leqslant b \end{array}\right\} \; b = y_j$$

Repeat for teams $n-1, \ldots, n-j+1$. Every true preference is now in place:

$$f(P_1, \ldots, P_n) = y_j = g(P).$$

Case 1: Walk the Right Teams In

Now swap team n from P_n^1 to its true P_n:

$$f(P_1, \ldots, P_{n-j}, P^1_{n-j+1}, \ldots, P_n) = b \quad \text{(say)}$$

$$\left.\begin{array}{l} y_j \; R_n^1 \; b \text{ and } P_n^1 \text{ peaks rightmost} \implies b \leqslant y_j \\ b \; R_n \; y_j \text{ and } b \leqslant y_j \leqslant p_n \implies y_j \leqslant b \end{array}\right\} \; b = y_j$$

Repeat for teams $n-1, \ldots, n-j+1$. Every true preference is now in place:

$$f(P_1, \ldots, P_n) = y_j = g(P).$$

In words

Each team's lie was pinned between two SP inequalities. Nobody could move the outcome. So the phantom y_j survives all the way to the true profile.



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

Case 2: The Median Is a Team's Peak

(≈ 25 minutes)

Case 2: Two Teams Are Enough to See It

Take $N = \{1, 2\}$. One phantom y_1. So $g(P) = \text{median}(p_1, p_2, y_1)$.

Case 2: Two Teams Are Enough to See It

Take $N = \{1, 2\}$. One phantom y_1. So $g(P) = \text{median}(p_1, p_2, y_1)$.

First, a lemma. It says only the peaks matter.

Lemma

If P and P' have the same peaks, $P_i(1) = P'_i(1)$ for $i = 1, 2$, then $f(P) = f(P')$.

Case 2: Two Teams Are Enough to See It

Take $N = \{1, 2\}$. One phantom y_1. So $g(P) = \text{median}(p_1, p_2, y_1)$.

First, a lemma. It says only the peaks matter.

Lemma

If P and P' have the same peaks, $P_i(1) = P'_i(1)$ for $i = 1, 2$, then $f(P) = f(P')$.

Proof. Let $a = P_1(1) = P'_1(1)$, $b = P_2(1) = P'_2(1)$. Let $x = f(P_1, P_2)$, $y = f(P'_1, P_2)$.

- WLOG $a < b$ (else pick Player 2 for the following arguments)

Case 2: Two Teams Are Enough to See It

Take $N = \{1, 2\}$. One phantom y_1. So $g(P) = \text{median}(p_1, p_2, y_1)$.

First, a lemma. It says only the peaks matter.

Lemma

If P and P' have the same peaks, $P_i(1) = P'_i(1)$ for $i = 1, 2$, then $f(P) = f(P')$.

Proof. Let $a = P_1(1) = P'_1(1)$, $b = P_2(1) = P'_2(1)$. Let $x = f(P_1, P_2)$, $y = f(P'_1, P_2)$.

- WLOG $a < b$ (else pick Player 2 for the following arguments)
- SP both ways: $x \; R_1 \; y$ and $y \; R'_1 \; x$.

Case 2: Two Teams Are Enough to See It

Take $N = \{1, 2\}$. One phantom y_1. So $g(P) = \text{median}(p_1, p_2, y_1)$.

First, a lemma. It says only the peaks matter.

Lemma

If P and P' have the same peaks, $P_i(1) = P'_i(1)$ for $i = 1, 2$, then $f(P) = f(P')$.

Proof. Let $a = P_1(1) = P'_1(1)$, $b = P_2(1) = P'_2(1)$. Let $x = f(P_1, P_2)$, $y = f(P'_1, P_2)$.

- WLOG $a < b$ (else pick Player 2 for the following arguments)
- SP both ways: $x \; R_1 \; y$ and $y \; R'_1 \; x$.
- If x, y are on the same side of a: both rankings prefer the closer one. So $x = y$.

Case 2: Two Teams Are Enough to See It

Take $N = \{1, 2\}$. One phantom y_1. So $g(P) = \text{median}(p_1, p_2, y_1)$.

First, a lemma. It says only the peaks matter.

Lemma

If P and P' have the same peaks, $P_i(1) = P'_i(1)$ for $i = 1, 2$, then $f(P) = f(P')$.

Proof. Let $a = P_1(1) = P'_1(1)$, $b = P_2(1) = P'_2(1)$. Let $x = f(P_1, P_2)$, $y = f(P'_1, P_2)$.

- WLOG $a < b$ (else pick Player 2 for the following arguments)
- SP both ways: $x \; R_1 \; y$ and $y \; R'_1 \; x$.
- If x, y are on the same side of a: both rankings prefer the closer one. So $x = y$.
- WLOG $x < a < y$. Recall SP + ONTO $\iff$ SP + PE. PE needs both outcomes in $[a, b]$. But $x < a$. Contradiction.

Case 2: Two Teams Are Enough to See It

Take $N = \{1, 2\}$. One phantom y_1. So $g(P) = \text{median}(p_1, p_2, y_1)$.

First, a lemma. It says only the peaks matter.

Lemma

If P and P' have the same peaks, $P_i(1) = P'_i(1)$ for $i = 1, 2$, then $f(P) = f(P')$.

Proof. Let $a = P_1(1) = P'_1(1)$, $b = P_2(1) = P'_2(1)$. Let $x = f(P_1, P_2)$, $y = f(P'_1, P_2)$.

- WLOG $a < b$ (else pick Player 2 for the following arguments)
- SP both ways: $x \; R_1 \; y$ and $y \; R'_1 \; x$.
- If x, y are on the same side of a: both rankings prefer the closer one. So $x = y$.
- WLOG $x < a < y$. Recall SP + ONTO $\iff$ SP + PE. PE needs both outcomes in $[a, b]$. But $x < a$. Contradiction.
- Same argument for $(P'_1, P_2) \to (P'_1, P'_2)$. □

Case 2: The Setup

Profile P: team 1 peaks at a, team 2 at b. WLOG the median of $\{a, b, y_1\}$ is a.

Case 2: The Setup

Profile P: team 1 peaks at a, team 2 at b. WLOG the median of $\{a, b, y_1\}$ is a.

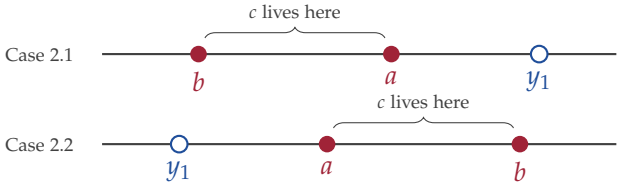
Suppose, for contradiction, $f(P) = c \neq a$.

Case 2: The Setup

Profile P: team 1 peaks at a, team 2 at b. WLOG the median of $\{a, b, y_1\}$ is a.

Suppose, for contradiction, $f(P) = c \neq a$.

By PE, c lies between a and b. The median is a, so there are two orderings:

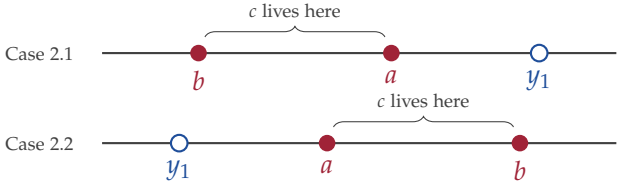


Case 2: The Setup

Profile P: team 1 peaks at a, team 2 at b. WLOG the median of $\{a, b, y_1\}$ is a.

Suppose, for contradiction, $f(P) = c \neq a$.

By PE, c lies between a and b. The median is a, so there are two orderings:



The trick: make team 1 *want* the phantom, then show it can grab it.

Case 2.1: $b < a < y_1$

By PE, $c < a$. So c and y_1 sit on opposite sides of a.

- Build P_1' with the same peak a, but ranking y_1 above c. Allowed, since they are on different sides of a.

Case 2.1: $b < a < y_1$

By PE, $c < a$. So c and y_1 sit on opposite sides of a.

- Build P_1' with the same peak a, but ranking y_1 above c. Allowed, since they are on different sides of a.
- Lemma: same peaks, same outcome. So $f(P_1', P_2) = c$.

Case 2.1: $b < a < y_1$

By PE, $c < a$. So c and y_1 sit on opposite sides of a.

- Build P_1' with the same peak a, but ranking y_1 above c. Allowed, since they are on different sides of a.
- Lemma: same peaks, same outcome. So $f(P_1', P_2) = c$.
- Now team 1 reports P_1^1, peak at the far right.

Case 2.1: $b < a < y_1$

By PE, $c < a$. So c and y_1 sit on opposite sides of a.

- Build P_1' with the same peak a, but ranking y_1 above c. Allowed, since they are on different sides of a.
- Lemma: same peaks, same outcome. So $f(P_1', P_2) = c$.
- Now team 1 reports P_1^1, peak at the far right.
- Median of $\{b,\ y_1,\ P_1^1(1)\}$ is y_1. That is a phantom. Case 1 applies: $f(P_1^1, P_2) = y_1$.

Case 2.1: $b < a < y_1$

By PE, $c < a$. So c and y_1 sit on opposite sides of a.

- Build P_1' with the same peak a, but ranking y_1 above c. Allowed, since they are on different sides of a.
- Lemma: same peaks, same outcome. So $f(P_1', P_2) = c$.
- Now team 1 reports P_1^1, peak at the far right.
- Median of $\{b,\ y_1,\ P_1^1(1)\}$ is y_1. That is a phantom. Case 1 applies: $f(P_1^1, P_2) = y_1$.
- Team 1 with true type P_1' gets c by telling the truth, y_1 by lying. It prefers y_1.

Case 2.1: $b < a < y_1$

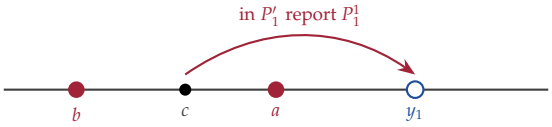
By PE, $c < a$. So c and y_1 sit on opposite sides of a.

- Build P_1' with the same peak a, but ranking y_1 above c. Allowed, since they are on different sides of a.
- Lemma: same peaks, same outcome. So $f(P_1', P_2) = c$.
- Now team 1 reports P_1^1, peak at the far right.
- Median of $\{b,\ y_1,\ P_1^1(1)\}$ is y_1. That is a phantom. Case 1 applies: $f(P_1^1, P_2) = y_1$.
- Team 1 with true type P_1' gets c by telling the truth, y_1 by lying. It prefers y_1.
- f is manipulable. Contradiction with SP.

Case 2.1: $b < a < y_1$

By PE, $c < a$. So c and y_1 sit on opposite sides of a.

- Build P_1' with the same peak a, but ranking y_1 above c. Allowed, since they are on different sides of a.
- Lemma: same peaks, same outcome. So $f(P_1', P_2) = c$.
- Now team 1 reports P_1^1, peak at the far right.
- Median of $\{b,\ y_1,\ P_1^1(1)\}$ is y_1. That is a phantom. Case 1 applies: $f(P_1^1, P_2) = y_1$.
- Team 1 with true type P_1' gets c by telling the truth, y_1 by lying. It prefers y_1.
- f is manipulable. Contradiction with SP.



Case 2.2: $y_1 < a < b$

Mirror image. By PE, $a < c$.

- Build P_1' with peak a that ranks y_1 above c. Lemma: $f(P_1', P_2) = c$.

Case 2.2: $y_1 < a < b$

Mirror image. By PE, $a < c$.

- Build P_1' with peak a that ranks y_1 above c. Lemma: $f(P_1', P_2) = c$.
- Team 1 reports P_1^0, peak at the far left.

Case 2.2: $y_1 < a < b$



Mirror image. By PE, $a < c$.

- Build P_1' with peak a that ranks y_1 above c. Lemma: $f(P_1', P_2) = c$.
- Team 1 reports P_1^0, peak at the far left.
- $P_1^0(1) \leqslant y_1 < b$. The median is the phantom y_1. Case 1: $f(P_1^0, P_2) = y_1$.

Case 2.2: $y_1 < a < b$

Mirror image. By PE, $a < c$.

- Build P_1' with peak a that ranks y_1 above c. Lemma: $f(P_1', P_2) = c$.
- Team 1 reports P_1^0, peak at the far left.
- $P_1^0(1) \leqslant y_1 < b$. The median is the phantom y_1. Case 1: $f(P_1^0, P_2) = y_1$.
- Team 1 with type P_1' gains by lying. Contradiction.

Case 2.2: $y_1 < a < b$

Mirror image. By PE, $a < c$.

- Build P_1' with peak a that ranks y_1 above c. Lemma: $f(P_1', P_2) = c$.
- Team 1 reports P_1^0, peak at the far left.
- $P_1^0(1) \leqslant y_1 < b$. The median is the phantom y_1. Case 1: $f(P_1^0, P_2) = y_1$.
- Team 1 with type P_1' gains by lying. Contradiction.

Wrapping up

So $f(P) = a = g(P)$ in Case 2 as well. For two teams, f *is* the median rule with phantom y_1. For general n, the same ideas go through with more bookkeeping. See **Moulin (1980), *On strategy-proofness and single peakedness***.

Case 2.2: $y_1 < a < b$

Mirror image. By PE, $a < c$.

- Build P_1' with peak a that ranks y_1 above c. Lemma: $f(P_1', P_2) = c$.
- Team 1 reports P_1^0, peak at the far left.
- $P_1^0(1) \leqslant y_1 < b$. The median is the phantom y_1. Case 1: $f(P_1^0, P_2) = y_1$.
- Team 1 with type P_1' gains by lying. Contradiction.

Wrapping up

So $f(P) = a = g(P)$ in Case 2 as well. For two teams, f *is* the median rule with phantom y_1. For general n, the same ideas go through with more bookkeeping. See **Moulin (1980), *On strategy-proofness and single peakedness***.

Big picture: single-peakedness + SP + ONTO + ANON leaves exactly one family of rules: **median voter**. GS said "only dictators". Here, a whole spectrum survives.



Chunk 4

Sharing a Divisible Task

(≈ 20 minutes)

A New Setting: Tonight's Delivery Load

A gig platform has **one unit** of delivery load tonight. It splits the load among n riders.

A New Setting: Tonight's Delivery Load

A gig platform has **one unit** of delivery load tonight. It splits the load among n riders.

- Rider i gets a share $s_i \in [0, 1]$, with $\sum_{i \in N} s_i = 1$.
- Every rider has a **most preferred share**.

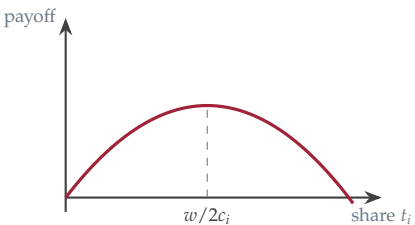
A New Setting: Tonight's Delivery Load

A gig platform has **one unit** of delivery load tonight. It splits the load among n riders.

- Rider i gets a share $s_i \in [0, 1]$, with $\sum_{i \in N} s_i = 1$.
- Every rider has a **most preferred share**.

Why a peak?

- Pay: w per unit of load. Working t_i hours earns $w\, t_i$.



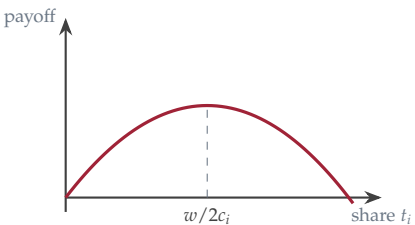
A New Setting: Tonight's Delivery Load

A gig platform has **one unit** of delivery load tonight. It splits the load among n riders.

- Rider i gets a share $s_i \in [0, 1]$, with $\sum_{i \in N} s_i = 1$.
- Every rider has a **most preferred share**.

Why a peak?

- Pay: w per unit of load. Working t_i hours earns $w\, t_i$.
- Fatigue: $c_i\, t_i^2$. Long working hours hurt (quadratically) more.



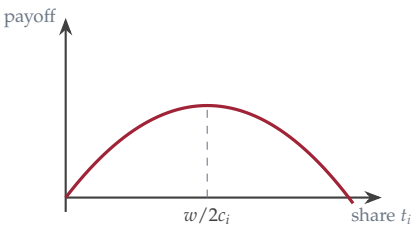
A New Setting: Tonight's Delivery Load

A gig platform has **one unit** of delivery load tonight. It splits the load among n riders.

- Rider i gets a share $s_i \in [0, 1]$, with $\sum_{i \in N} s_i = 1$.
- Every rider has a **most preferred share**.

Why a peak?

- Pay: w per unit of load. Working t_i hours earns $w\, t_i$.
- Fatigue: $c_i\, t_i^2$. Long working hours hurt (quadratically) more.
- Net: $w\, t_i - c_i\, t_i^2$. Maximized at $t_i = w/2c_i$.



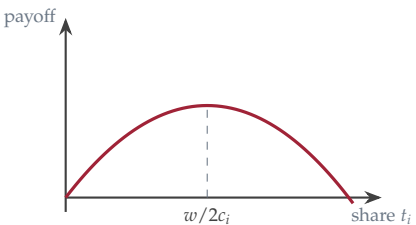
A New Setting: Tonight's Delivery Load

A gig platform has **one unit** of delivery load tonight. It splits the load among n riders.

- Rider i gets a share $s_i \in [0, 1]$, with $\sum_{i \in N} s_i = 1$.
- Every rider has a **most preferred share**.

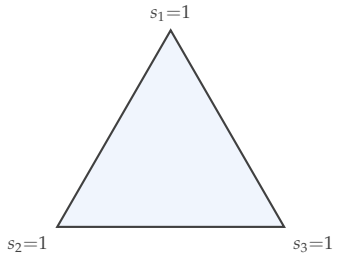
Why a peak?

- Pay: w per unit of load. Working t_i hours earns $w\, t_i$.
- Fatigue: $c_i\, t_i^2$. Long working hours hurt (quadratically) more.
- Net: $w\, t_i - c_i\, t_i^2$. Maximized at $t_i = w/2c_i$.
- It falls on both sides. **Single-peaked** in the share.



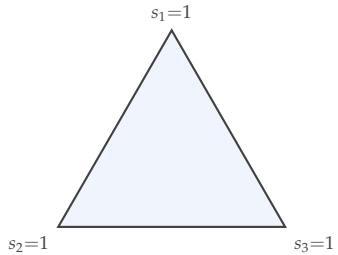
Single-Peaked Over *Shares*, Not Over Allocations

- An alternative is now a whole **allocation** $(s_1, \ldots, s_n)$.



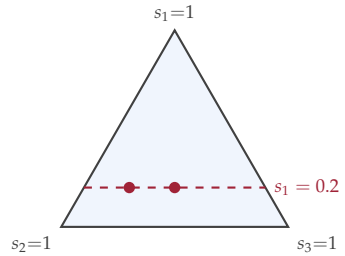
Single-Peaked Over *Shares*, Not Over Allocations

- An alternative is now a whole **allocation** $(s_1, \ldots, s_n)$.
- Rider 1 only looks at s_1.



Single-Peaked Over *Shares*, Not Over Allocations

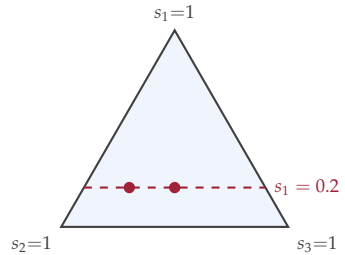
- An alternative is now a whole **allocation** $(s_1, \ldots, s_n)$.
- Rider 1 only looks at s_1.
- $(0.2, 0.4, 0.4)$ and $(0.2, 0.6, 0.2)$: rider 1 likes both *equally*.



every point on the dashed line looks the same to rider 1

Single-Peaked Over *Shares*, Not Over Allocations

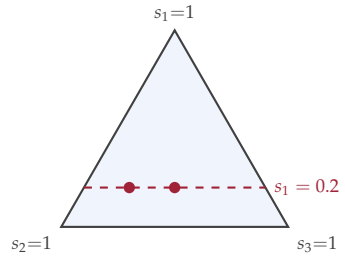
- An alternative is now a whole **allocation** $(s_1, \ldots, s_n)$.
- Rider 1 only looks at s_1.
- $(0.2, 0.4, 0.4)$ and $(0.2, 0.6, 0.2)$: rider 1 likes both *equally*.
- For 3 riders, the alternatives form a triangle (a simplex).



every point on the dashed line looks the same to rider 1

Single-Peaked Over *Shares*, Not Over Allocations

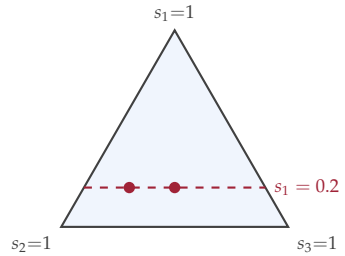
- An alternative is now a whole **allocation** $(s_1, \ldots, s_n)$.
- Rider 1 only looks at s_1.
- $(0.2, 0.4, 0.4)$ and $(0.2, 0.6, 0.2)$: rider 1 likes both *equally*.
- For 3 riders, the alternatives form a triangle (a simplex).
- No single common order makes everyone single-peaked over allocations.



every point on the dashed line looks the same to rider 1

Single-Peaked Over *Shares*, Not Over Allocations

- An alternative is now a whole **allocation** $(s_1, \ldots, s_n)$.
- Rider 1 only looks at s_1.
- $(0.2, 0.4, 0.4)$ and $(0.2, 0.6, 0.2)$: rider 1 likes both *equally*.
- For 3 riders, the alternatives form a triangle (a simplex).
- No single common order makes everyone single-peaked over allocations.
- So Moulin's theorem does not apply directly. We need new rules.



every point on the dashed line looks the same to rider 1

SCFs and Pareto Efficiency for Shares

- The **domain of task allocation**: T. Allocations: A.
- **SCF**: $f : T^n \to A$, with $f(P) = (f_1(P), \ldots, f_n(P))$, $f_i(P) \in [0, 1]$, $\sum_i f_i(P) = 1$.
- Rider i's peak share is p_i.

SCFs and Pareto Efficiency for Shares

- The **domain of task allocation**: T. Allocations: A.
- **SCF**: $f : T^n \to A$, with $f(P) = (f_1(P), \ldots, f_n(P))$, $f_i(P) \in [0,1]$, $\sum_i f_i(P) = 1$.
- Rider i's peak share is p_i.

Definition (Pareto Efficiency)

f is *Pareto efficient* (PE) if there is no profile P and allocation $a \in A$ such that

$$a \; R_i \, f(P) \;\; \forall i \in N, \qquad a \; P_j \, f(P) \;\; \text{for some } j \in N.$$

SCFs and Pareto Efficiency for Shares

- The **domain of task allocation**: T. Allocations: A.
- **SCF**: $f : T^n \to A$, with $f(P) = (f_1(P), \ldots, f_n(P))$, $f_i(P) \in [0,1]$, $\sum_i f_i(P) = 1$.
- Rider i's peak share is p_i.

Definition (Pareto Efficiency)

f is *Pareto efficient* (PE) if there is no profile P and allocation $a \in A$ such that

$$a \; R_i \, f(P) \;\; \forall i \in N, \qquad a \; P_j \, f(P) \;\; \text{for some } j \in N.$$

In words: you cannot make one rider happier without making another rider sadder.

What PE Forces

1. $\sum_i p_i = 1$: give everyone exactly p_i. This is the *unique* PE allocation.

What PE Forces

1. $\sum_i p_i = 1$: give everyone exactly p_i. This is the *unique* PE allocation.
2. $\sum_i p_i > 1$: too little work to go around. Some rider k must get $f_k(P) < p_k$.

What PE Forces

1. $\sum_i p_i = 1$: give everyone exactly p_i. This is the *unique* PE allocation.
2. $\sum_i p_i > 1$: too little work to go around. Some rider k must get $f_k(P) < p_k$.
3. $\sum_i p_i < 1$: too much work. By the mirror argument, $f_k(P) > p_k$ for some k.

What PE Forces

1. $\sum_i p_i = 1$: give everyone exactly p_i. This is the *unique* PE allocation.
2. $\sum_i p_i > 1$: too little work to go around. Some rider k must get $f_k(P) < p_k$.
3. $\sum_i p_i < 1$: too much work. By the mirror argument, $f_k(P) > p_k$ for some k.

Question

In case 2, can some rider j get $f_j(P) > p_j$ if f is PE?

What PE Forces

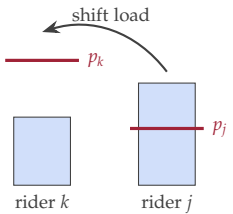
1. $\sum_i p_i = 1$: give everyone exactly p_i. This is the *unique* PE allocation.
2. $\sum_i p_i > 1$: too little work to go around. Some rider k must get $f_k(P) < p_k$.
3. $\sum_i p_i < 1$: too much work. By the mirror argument, $f_k(P) > p_k$ for some k.

Question

In case 2, can some rider j get $f_j(P) > p_j$ if f is PE?

Answer

No. Move a sliver of load from j to k. Rider j wanted less, rider k wanted more. Both strictly gain. So $f_j(P) \leqslant p_j$ for every j.



What PE Forces

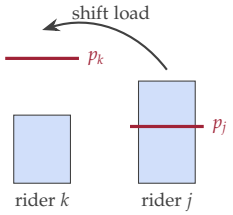
1. $\sum_i p_i = 1$: give everyone exactly p_i. This is the *unique* PE allocation.
2. $\sum_i p_i > 1$: too little work to go around. Some rider k must get $f_k(P) < p_k$.
3. $\sum_i p_i < 1$: too much work. By the mirror argument, $f_k(P) > p_k$ for some k.

Question

In case 2, can some rider j get $f_j(P) > p_j$ if f is PE?

Answer

No. Move a sliver of load from j to k. Rider j wanted less, rider k wanted more. Both strictly gain. So $f_j(P) \leqslant p_j$ for every j.



Rule of thumb

PE $\iff$ every rider stays on the *same side* of her peak.

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 17: What We Built

- Phantom peaks come from f itself: y_j is the outcome when j teams go far right.

Lecture 17: What We Built

- Phantom peaks come from f itself: y_j is the outcome when j teams go far right.
- SP forces them to be sorted.

Lecture 17: What We Built

- Phantom peaks come from f itself: y_j is the outcome when j teams go far right.
- SP forces them to be sorted.
- Case 1: two SP inequalities pin each team's switch. The phantom median survives.

Lecture 17: What We Built

- Phantom peaks come from f itself: y_j is the outcome when j teams go far right.
- SP forces them to be sorted.
- Case 1: two SP inequalities pin each team's switch. The phantom median survives.
- Case 2: a team that could steer to the phantom would lie. So the team median must win.

Lecture 17: What We Built

- Phantom peaks come from f itself: y_j is the outcome when j teams go far right.
- SP forces them to be sorted.
- Case 1: two SP inequalities pin each team's switch. The phantom median survives.
- Case 2: a team that could steer to the phantom would lie. So the team median must win.
- Moulin: on single-peaked domains, SP + ONTO + ANON = median voter rules.

Lecture 17: What We Built

- Phantom peaks come from f itself: y_j is the outcome when j teams go far right.
- SP forces them to be sorted.
- Case 1: two SP inequalities pin each team's switch. The phantom median survives.
- Case 2: a team that could steer to the phantom would lie. So the team median must win.
- Moulin: on single-peaked domains, SP + ONTO + ANON = median voter rules.
- Task sharing: single-peaked in one's own share. PE means same side of the peak.

Lecture 17: What We Built

- Phantom peaks come from f itself: y_j is the outcome when j teams go far right.
- SP forces them to be sorted.
- Case 1: two SP inequalities pin each team's switch. The phantom median survives.
- Case 2: a team that could steer to the phantom would lie. So the team median must win.
- Moulin: on single-peaked domains, SP + ONTO + ANON = median voter rules.
- Task sharing: single-peaked in one's own share. PE means same side of the peak.
- **Next lecture:** which rule is PE, ANON, and SP for shares?



End of Lecture 17

See you next class.

Contents

- Lecture 17: The Median Voter Theorem and Task Sharing
- Lecture 18: The Uniform Rule and Mechanisms with Money



Chunk 1

Fair and Truthful Splits

(≈ 20 minutes)

Anonymity for Shares



Renaming riders should just rename their shares.

Anonymity for Shares

Renaming riders should just rename their shares.

Definition (Anonymity)

An SCF f is *anonymous* (ANON) if for every agent permutation $\sigma : N \to N$, the task shares get permuted accordingly, i.e.,

$$\forall \sigma,\ f_{\sigma(j)}(P^{\sigma}) = f_j(P),\ \forall j \in N.$$

Anonymity for Shares

Renaming riders should just rename their shares.

Definition (Anonymity)

An SCF f is *anonymous* (ANON) if for every agent permutation $\sigma : N \to N$, the task shares get permuted accordingly, i.e.,

$$\forall \sigma,\ f_{\sigma(j)}(P^\sigma) = f_j(P),\ \forall j \in N.$$

Example: three riders, $N = \{1, 2, 3\}$, $\sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1$.

- $P = (0.7, 0.4, 0.3) \implies P^\sigma = (0.3, 0.7, 0.4)$

Anonymity for Shares

Renaming riders should just rename their shares.

Definition (Anonymity)

An SCF f is *anonymous* (ANON) if for every agent permutation $\sigma : N \to N$, the task shares get permuted accordingly, i.e.,

$$\forall \sigma,\ f_{\sigma(j)}(P^\sigma) = f_j(P),\ \forall j \in N.$$

Example: three riders, $N = \{1,2,3\}$, $\sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1$.

- $P = (0.7, 0.4, 0.3) \implies P^\sigma = (0.3, 0.7, 0.4)$
- $f_1(0.7, 0.4, 0.3) = f_2(0.3, 0.7, 0.4)$
- $f_2(0.7, 0.4, 0.3) = f_3(0.3, 0.7, 0.4)$
- $f_3(0.7, 0.4, 0.3) = f_1(0.3, 0.7, 0.4)$

Anonymity for Shares

Renaming riders should just rename their shares.

Definition (Anonymity)

An SCF f is *anonymous* (ANON) if for every agent permutation $\sigma : N \to N$, the task shares get permuted accordingly, i.e.,

$$\forall \sigma,\ f_{\sigma(j)}(P^\sigma) = f_j(P),\ \forall j \in N.$$

Example: three riders, $N = \{1,2,3\}$, $\sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1$.

- $P = (0.7, 0.4, 0.3) \implies P^\sigma = (0.3, 0.7, 0.4)$
- $f_1(0.7, 0.4, 0.3) = f_2(0.3, 0.7, 0.4)$
- $f_2(0.7, 0.4, 0.3) = f_3(0.3, 0.7, 0.4)$
- $f_3(0.7, 0.4, 0.3) = f_1(0.3, 0.7, 0.4)$

The rider who asked for 0.7 gets the same load, whatever her name.

Strategyproofness for Shares

Manipulability: f is **manipulable** if $\exists i \in N$, a profile P, and a report P_i' with

$$f(P_i', P_{-i}) \; P_i \, f(P_i, P_{-i}).$$

Strategyproofness for Shares

Manipulability: f is **manipulable** if $\exists i \in N$, a profile P, and a report P_i' with

$$f(P_i', P_{-i}) \; P_i \, f(P_i, P_{-i}).$$

Definition (Strategyproof)

An SCF is *strategyproof* (SP) if it is not manipulable by any agent at any profile.

Strategyproofness for Shares

Manipulability: f is **manipulable** if $\exists i \in N$, a profile P, and a report P_i' with

$$f(P_i', P_{-i}) \; P_i \, f(P_i, P_{-i}).$$

Definition (Strategyproof)

An SCF is *strategyproof* (SP) if it is not manipulable by any agent at any profile.

Equivalent form. Only rider i's own share matters to her:

$$f(P_i, P_{-i}) \; P_i \, f(P_i', P_{-i}) \quad \text{OR} \quad f_i(P_i, P_{-i}) = f_i(P_i', P_{-i}),$$

$$\forall P_i, P_i' \in T, \; \forall i \in N, \; \forall P_{-i} \in T^{n-1}.$$

Candidate 1: Serial Dictatorship

Definition (Serial Dictatorship)

Fix an order of the riders in advance. Each rider in turn gets her peak share, or whatever is left. If $\sum_{i \in N} p_i < 1$, the last rider takes the leftover.

Candidate 1: Serial Dictatorship

Definition (Serial Dictatorship)

Fix an order of the riders in advance. Each rider in turn gets her peak share, or whatever is left. If $\sum_{i \in N} p_i < 1$, the last rider takes the leftover.

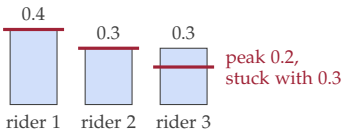
Example: peaks $p_1 = 0.4$, $p_2 = 0.3$, $p_3 = 0.2$, order $1, 2, 3$.

Candidate 1: Serial Dictatorship

Definition (Serial Dictatorship)

Fix an order of the riders in advance. Each rider in turn gets her peak share, or whatever is left. If $\sum_{i \in N} p_i < 1$, the last rider takes the leftover.

Example: peaks $p_1 = 0.4$, $p_2 = 0.3$, $p_3 = 0.2$, order $1, 2, 3$.



Question

PE, SP, ANON?

Candidate 1: Serial Dictatorship

Definition (Serial Dictatorship)

Fix an order of the riders in advance. Each rider in turn gets her peak share, or whatever is left. If $\sum_{i \in N} p_i < 1$, the last rider takes the leftover.

Example: peaks $p_1 = 0.4$, $p_2 = 0.3$, $p_3 = 0.2$, order $1, 2, 3$.



Question

PE, SP, ANON?

Answer

PE and SP: early riders get their peak, later ones cannot change what is left. Not ANON: the order decides who is stuck. Quite unfair to the last rider.

Candidate 2: Proportional

Definition (Proportional)

Every rider gets c times her peak, where $c \sum_{i \in N} p_i = 1$.

Candidate 2: Proportional

Definition (Proportional)

Every rider gets c times her peak, where $c \sum_{i \in N} p_i = 1$.

Question

PE, ANON, SP?

Candidate 2: Proportional

Definition (Proportional)

Every rider gets c times her peak, where $c \sum_{i \in N} p_i = 1$.

Question

PE, ANON, SP?

Answer

PE and ANON. **Not SP.**
Peaks are $0.2, 0.3, 0.1$. So $c = 1/0.6$.

Candidate 2: Proportional

Definition (Proportional)

Every rider gets c times her peak, where $c \sum_{i \in N} p_i = 1$.

Question

PE, ANON, SP?

Answer

PE and ANON. **Not SP.**
Peaks are $0.2, 0.3, 0.1$. So $c = 1/0.6$.
Rider 1 gets $1/3$. That is more than her peak 0.2.

Candidate 2: Proportional

Definition (Proportional)

Every rider gets c times her peak, where $c \sum_{i \in N} p_i = 1$.

Question

PE, ANON, SP?

Answer

PE and ANON. **Not SP.**
Peaks are $0.2, 0.3, 0.1$. So $c = 1/0.6$.
Rider 1 gets $1/3$. That is more than her peak 0.2.
She reports $0.1, 0.3, 0.1$ instead. Now $c = 1/0.5$, and she gets exactly 0.2.

Candidate 2: Proportional

Definition (Proportional)

Every rider gets c times her peak, where $c \sum_{i \in N} p_i = 1$.

Question

PE, ANON, SP?

Answer

PE and ANON. **Not SP.**
Peaks are $0.2, 0.3, 0.1$. So $c = 1/0.6$.
Rider 1 gets $1/3$. That is more than her peak 0.2.
She reports $0.1, 0.3, 0.1$ instead. Now $c = 1/0.5$, and she gets exactly 0.2.

The pattern

If $\sum_j p_j < 1$, under-report to shed load. If $\sum_j p_j > 1$, over-report to grab load.



Chunk 2

The Uniform Rule

(≈ 25 minutes)

PE, ANON, and SP Together?

Question

How to ensure PE, ANON, and SP in the task allocation domain?

PE, ANON, and SP Together?

Question

How to ensure PE, ANON, and SP in the task allocation domain?

Take the anonymity example: peaks $(0.7, 0.4, 0.3)$. The sum is $1.4 > 1$.

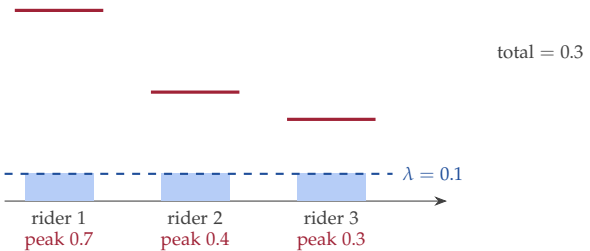
PE, ANON, and SP Together?

Question

How to ensure PE, ANON, and SP in the task allocation domain?

Take the anonymity example: peaks $(0.7, 0.4, 0.3)$. The sum is $1.4 > 1$.

Start every share at 0. Raise a common **water level** λ. Freeze a rider once she hits her peak. Stop when shares add to 1.



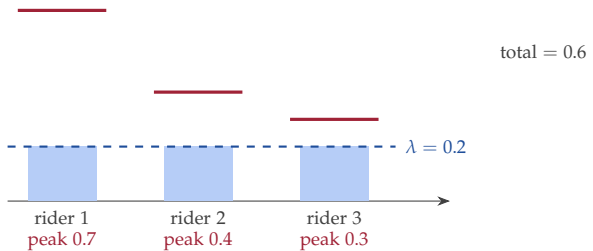
PE, ANON, and SP Together?

Question

How to ensure PE, ANON, and SP in the task allocation domain?

Take the anonymity example: peaks $(0.7, 0.4, 0.3)$. The sum is $1.4 > 1$.

Start every share at 0. Raise a common **water level** λ. Freeze a rider once she hits her peak. Stop when shares add to 1.



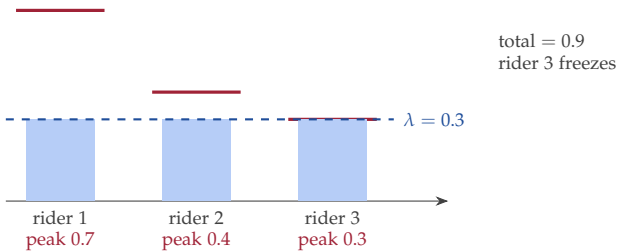
PE, ANON, and SP Together?

Question

How to ensure PE, ANON, and SP in the task allocation domain?

Take the anonymity example: peaks $(0.7, 0.4, 0.3)$. The sum is $1.4 > 1$.

Start every share at 0. Raise a common **water level** λ. Freeze a rider once she hits her peak. Stop when shares add to 1.



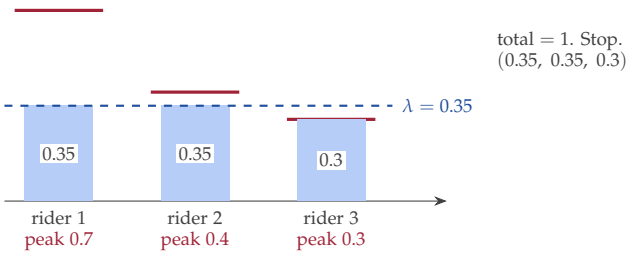
PE, ANON, and SP Together?

Question

How to ensure PE, ANON, and SP in the task allocation domain?

Take the anonymity example: peaks $(0.7, 0.4, 0.3)$. The sum is $1.4 > 1$.

Start every share at 0. Raise a common **water level** λ. Freeze a rider once she hits her peak. Stop when shares add to 1.



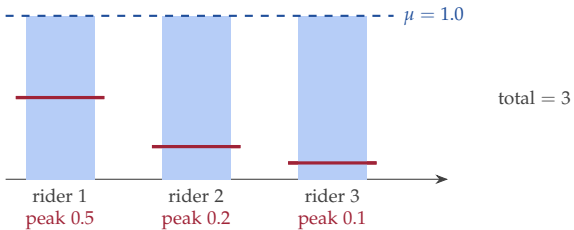
The Other Direction: Too Much Work

Now take peaks $(0.5, 0.2, 0.1)$. The sum is $0.8 < 1$.

The Other Direction: Too Much Work

Now take peaks $(0.5, 0.2, 0.1)$. The sum is $0.8 < 1$.

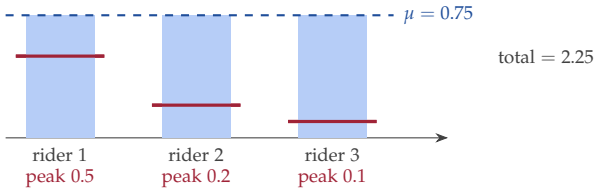
Start every share at 1 (infeasible). Lower a common **water level** μ. Freeze a rider once the level hits her peak. Stop when shares add to 1.



The Other Direction: Too Much Work

Now take peaks $(0.5, 0.2, 0.1)$. The sum is $0.8 < 1$.

Start every share at 1 (infeasible). Lower a common **water level** μ. Freeze a rider once the level hits her peak. Stop when shares add to 1.



The Other Direction: Too Much Work

Now take peaks $(0.5, 0.2, 0.1)$. The sum is $0.8 < 1$.

Start every share at 1 (infeasible). Lower a common **water level** μ. Freeze a rider once the level hits her peak. Stop when shares add to 1.



total $= 1.5$
rider 1 freezes

The Other Direction: Too Much Work

Now take peaks $(0.5, 0.2, 0.1)$. The sum is $0.8 < 1$.

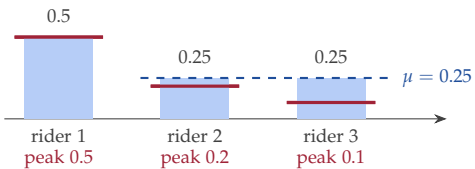
Start every share at 1 (infeasible). Lower a common **water level** μ. Freeze a rider once the level hits her peak. Stop when shares add to 1.



The Other Direction: Too Much Work

Now take peaks $(0.5, 0.2, 0.1)$. The sum is $0.8 < 1$.

Start every share at 1 (infeasible). Lower a common **water level** μ. Freeze a rider once the level hits her peak. Stop when shares add to 1.

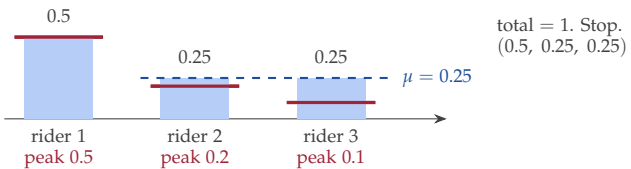


total $= 1$. Stop.
$(0.5,\ 0.25,\ 0.25)$

The Other Direction: Too Much Work

Now take peaks $(0.5, 0.2, 0.1)$. The sum is $0.8 < 1$.

Start every share at 1 (infeasible). Lower a common **water level** μ. Freeze a rider once the level hits her peak. Stop when shares add to 1.



total $= 1$. Stop.
$(0.5,\ 0.25,\ 0.25)$

Proportional would give $(0.625,\ 0.25,\ 0.125)$. Rider 1 would be pushed above her peak. Here she gets exactly 0.5.

The Uniform Rule (Sprumont 1991)



Definition (Uniform rule)

1. **Case** $\sum_{i \in N} p_i = 1$: $f_i^{\mu}(P) = p_i$

The Uniform Rule (Sprumont 1991)

Definition (Uniform rule)

1. **Case** $\sum_{i \in N} p_i = 1$: $f_i^{\mu}(P) = p_i$
2. **Case** $\sum_{i \in N} p_i < 1$: $f_i^{\mu}(P) = \max\{p_i, \mu(P)\}$, where $\mu(P)$ solves $\sum_{i \in N} \max\{p_i, \mu\} = 1$

The Uniform Rule (Sprumont 1991)

Definition (Uniform rule)

1. **Case** $\sum_{i \in N} p_i = 1$: $f_i^{\mu}(P) = p_i$
2. **Case** $\sum_{i \in N} p_i < 1$: $f_i^{\mu}(P) = \max\{p_i, \mu(P)\}$, where $\mu(P)$ solves $\sum_{i \in N} \max\{p_i, \mu\} = 1$
3. **Case** $\sum_{i \in N} p_i > 1$: $f_i^{\mu}(P) = \min\{p_i, \lambda(P)\}$, where $\lambda(P)$ solves $\sum_{i \in N} \min\{p_i, \lambda\} = 1$

The Uniform Rule (Sprumont 1991)

Definition (Uniform rule)

1. **Case** $\sum_{i \in N} p_i = 1$: $f_i^{\mu}(P) = p_i$
2. **Case** $\sum_{i \in N} p_i < 1$: $f_i^{\mu}(P) = \max\{p_i, \mu(P)\}$, where $\mu(P)$ solves $\sum_{i \in N} \max\{p_i, \mu\} = 1$
3. **Case** $\sum_{i \in N} p_i > 1$: $f_i^{\mu}(P) = \min\{p_i, \lambda(P)\}$, where $\lambda(P)$ solves $\sum_{i \in N} \min\{p_i, \lambda\} = 1$

Case $\sum_i p_i \leqslant 1$: $\max_{x, \mu} \ \mu \ \text{ s.t. } \ x_i \geqslant p_i, \ x_i \geqslant \mu \ \ \forall i, \ \ \sum_i x_i = 1$

Case $\sum_i p_i \geqslant 1$: $\min_{x, \lambda} \ \lambda \ \text{ s.t. } \ x_i \leqslant p_i, \ x_i \leqslant \lambda \ \ \forall i, \ \ \sum_i x_i = 1$

The Uniform Rule (Sprumont 1991)

Definition (Uniform rule)

1. **Case** $\sum_{i \in N} p_i = 1$: $f_i^{\mu}(P) = p_i$
2. **Case** $\sum_{i \in N} p_i < 1$: $f_i^{\mu}(P) = \max\{p_i, \mu(P)\}$, where $\mu(P)$ solves $\sum_{i \in N} \max\{p_i, \mu\} = 1$
3. **Case** $\sum_{i \in N} p_i > 1$: $f_i^{\mu}(P) = \min\{p_i, \lambda(P)\}$, where $\lambda(P)$ solves $\sum_{i \in N} \min\{p_i, \lambda\} = 1$

Case $\sum_i p_i \leqslant 1$: $\max_{x, \mu} \ \mu$ s.t. $x_i \geqslant p_i,\ x_i \geqslant \mu \ \ \forall i, \ \ \sum_i x_i = 1$

Case $\sum_i p_i \geqslant 1$: $\min_{x, \lambda} \ \lambda$ s.t. $x_i \leqslant p_i,\ x_i \leqslant \lambda \ \ \forall i, \ \ \sum_i x_i = 1$

Theorem (Sprumont 1991)

*The **uniform rule** SCF is ANON, PE, and SP.*

The Uniform Rule (Sprumont 1991)

Definition (Uniform rule)

1. **Case** $\sum_{i \in N} p_i = 1$: $f_i^{\mu}(P) = p_i$
2. **Case** $\sum_{i \in N} p_i < 1$: $f_i^{\mu}(P) = \max\{p_i, \mu(P)\}$, where $\mu(P)$ solves $\sum_{i \in N} \max\{p_i, \mu\} = 1$
3. **Case** $\sum_{i \in N} p_i > 1$: $f_i^{\mu}(P) = \min\{p_i, \lambda(P)\}$, where $\lambda(P)$ solves $\sum_{i \in N} \min\{p_i, \lambda\} = 1$

Case $\sum_i p_i \leqslant 1$: $\max_{x, \mu} \ \mu$ s.t. $x_i \geqslant p_i$, $x_i \geqslant \mu \ \ \forall i$, $\sum_i x_i = 1$

Case $\sum_i p_i \geqslant 1$: $\min_{x, \lambda} \ \lambda$ s.t. $x_i \leqslant p_i$, $x_i \leqslant \lambda \ \ \forall i$, $\sum_i x_i = 1$

Theorem (Sprumont 1991)

*The **uniform rule** SCF is ANON, PE, and SP.*

- **ANON** is obvious: only the peaks matter, not their owners.

The Uniform Rule (Sprumont 1991)

Definition (Uniform rule)

1. **Case** $\sum_{i \in N} p_i = 1$: $f_i^{\mu}(P) = p_i$
2. **Case** $\sum_{i \in N} p_i < 1$: $f_i^{\mu}(P) = \max\{p_i, \mu(P)\}$, where $\mu(P)$ solves $\sum_{i \in N} \max\{p_i, \mu\} = 1$
3. **Case** $\sum_{i \in N} p_i > 1$: $f_i^{\mu}(P) = \min\{p_i, \lambda(P)\}$, where $\lambda(P)$ solves $\sum_{i \in N} \min\{p_i, \lambda\} = 1$

Case $\sum_i p_i \leqslant 1$: $\max_{x, \mu} \ \mu$ s.t. $x_i \geqslant p_i,\ x_i \geqslant \mu \ \ \forall i, \ \ \sum_i x_i = 1$

Case $\sum_i p_i \geqslant 1$: $\min_{x, \lambda} \ \lambda$ s.t. $x_i \leqslant p_i,\ x_i \leqslant \lambda \ \ \forall i, \ \ \sum_i x_i = 1$

Theorem (Sprumont 1991)

*The **uniform rule** SCF is ANON, PE, and SP.*

- **ANON** is obvious: only the peaks matter, not their owners.
- **PE**: in each case *everyone* gets $= p_i$, $\geqslant p_i$, or $\leqslant p_i$. Same side of the peak, so PE.

The Uniform Rule: Strategyproofness



- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.

The Uniform Rule: Strategyproofness

- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.
- **Case** $\sum_i p_i < 1$: then $f_i^u(P) \geqslant p_i$ for all i.

The Uniform Rule: Strategyproofness



- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.
- **Case** $\sum_i p_i < 1$: then $f_i^{\mu}(P) \geqslant p_i$ for all i.
- Only a rider with $f_i^{\mu}(P) > p_i$ might want to lie. For her, $f_i^{\mu}(P) = \mu(P) > p_i$.

The Uniform Rule: Strategyproofness

- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.
- **Case** $\sum_i p_i < 1$: then $f_i^{\mu}(P) \geqslant p_i$ for all i.
- Only a rider with $f_i^{\mu}(P) > p_i$ might want to lie. For her, $f_i^{\mu}(P) = \mu(P) > p_i$.
- Any report $p_i' \leqslant \mu(P)$ leaves μ, and her share, unchanged.

The Uniform Rule: Strategyproofness

- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.
- **Case** $\sum_i p_i < 1$: then $f_i^{\mu}(P) \geqslant p_i$ for all i.
- Only a rider with $f_i^{\mu}(P) > p_i$ might want to lie. For her, $f_i^{\mu}(P) = \mu(P) > p_i$.
- Any report $p_i' \leqslant \mu(P)$ leaves μ, and her share, unchanged.
- The only way to change the outcome is to report $p_i' > \mu(P) > p_i$.

The Uniform Rule: Strategyproofness

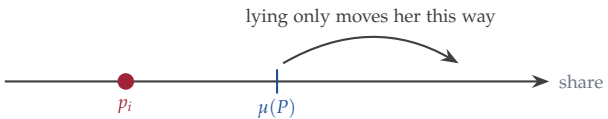
- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.
- **Case** $\sum_i p_i < 1$: then $f_i^{\mu}(P) \geqslant p_i$ for all i.
- Only a rider with $f_i^{\mu}(P) > p_i$ might want to lie. For her, $f_i^{\mu}(P) = \mu(P) > p_i$.
- Any report $p_i' \leqslant \mu(P)$ leaves μ, and her share, unchanged.
- The only way to change the outcome is to report $p_i' > \mu(P) > p_i$.
- That gives her *more* than $\mu(P)$. Even further from p_i. Worse.

The Uniform Rule: Strategyproofness

- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.
- **Case** $\sum_i p_i < 1$: then $f_i^\mu(P) \geqslant p_i$ for all i.
- Only a rider with $f_i^\mu(P) > p_i$ might want to lie. For her, $f_i^\mu(P) = \mu(P) > p_i$.
- Any report $p_i' \leqslant \mu(P)$ leaves μ, and her share, unchanged.
- The only way to change the outcome is to report $p_i' > \mu(P) > p_i$.
- That gives her *more* than $\mu(P)$. Even further from p_i. Worse.
- **Case** $\sum_i p_i > 1$: the mirror argument with λ.

The Uniform Rule: Strategyproofness

- **Case** $\sum_i p_i = 1$: everyone gets her peak. No reason to lie.
- **Case** $\sum_i p_i < 1$: then $f_i^{\mu}(P) \geqslant p_i$ for all i.
- Only a rider with $f_i^{\mu}(P) > p_i$ might want to lie. For her, $f_i^{\mu}(P) = \mu(P) > p_i$.
- Any report $p_i' \leqslant \mu(P)$ leaves μ, and her share, unchanged.
- The only way to change the outcome is to report $p_i' > \mu(P) > p_i$.
- That gives her *more* than $\mu(P)$. Even further from p_i. Worse.
- **Case** $\sum_i p_i > 1$: the mirror argument with λ.



The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

See **Sprumont (1991): The division problem with single-peaked preferences**.

The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

See **Sprumont (1991): The division problem with single-peaked preferences**.

Two more gifts.

- **Envy-free (EF):** $u_i(f_i(P)) \geqslant u_i(f_j(P))$ for all $i \neq j$.

The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

See **Sprumont (1991): The division problem with single-peaked preferences**.

Two more gifts.

- **Envy-free (EF):** $u_i(f_i(P)) \geqslant u_i(f_j(P))$ for all $i \neq j$.
  - Say $\sum_i p_i < 1$. Rider i could envy only if $f_i^u(P) = \mu(P) > p_i$.

The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

See **Sprumont (1991): The division problem with single-peaked preferences**.

Two more gifts.

- **Envy-free (EF):** $u_i(f_i(P)) \geqslant u_i(f_j(P))$ for all $i \neq j$.
  - Say $\sum_i p_i < 1$. Rider i could envy only if $f_i^u(P) = \mu(P) > p_i$.
  - Any other rider j gets $\max\{\mu(P), p_j\} \geqslant \mu(P)$. Even further from p_i.

The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

See **Sprumont (1991): The division problem with single-peaked preferences**.

Two more gifts.

- **Envy-free (EF):** $u_i(f_i(P)) \geqslant u_i(f_j(P))$ for all $i \neq j$.
  - Say $\sum_i p_i < 1$. Rider i could envy only if $f_i^u(P) = \mu(P) > p_i$.
  - Any other rider j gets $\max\{\mu(P), p_j\} \geqslant \mu(P)$. Even further from p_i.
  - So i never prefers j's share. The case $\sum_i p_i > 1$ is symmetric.

The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

See **Sprumont (1991): The division problem with single-peaked preferences**.

Two more gifts.

- **Envy-free (EF):** $u_i(f_i(P)) \geqslant u_i(f_j(P))$ for all $i \neq j$.
  - Say $\sum_i p_i < 1$. Rider i could envy only if $f_i^u(P) = \mu(P) > p_i$.
  - Any other rider j gets $\max\{\mu(P), p_j\} \geqslant \mu(P)$. Even further from p_i.
  - So i never prefers j's share. The case $\sum_i p_i > 1$ is symmetric.
- **Polynomial time.** Check the cases. Solve the small LP.

The Uniform Rule Is the Only One

Theorem (Sprumont 1991)

An SCF in the task allocation domain is SP, PE, and ANON iff it is the uniform rule.

See **Sprumont (1991): The division problem with single-peaked preferences**.

Two more gifts.

- **Envy-free (EF):** $u_i(f_i(P)) \geqslant u_i(f_j(P))$ for all $i \neq j$.
  - Say $\sum_i p_i < 1$. Rider i could envy only if $f_i^u(P) = \mu(P) > p_i$.
  - Any other rider j gets $\max\{\mu(P), p_j\} \geqslant \mu(P)$. Even further from p_i.
  - So i never prefers j's share. The case $\sum_i p_i > 1$ is symmetric.
- **Polynomial time.** Check the cases. Solve the small LP.

Uniform rule: SP, PE, ANON, EF, and fast.

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

Money Enters: Quasi-Linear Preferences

(≈ 20 minutes)

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:
  - an **allocation** $a \in A$

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:
  - an **allocation** $a \in A$
  - a **payment** vector $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$

Mechanism Design with Transfers



- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:
  - an **allocation** $a \in A$
  - a **payment** vector $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:
  - an **allocation** $a \in A$
  - a **payment** vector $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. **Public decision.** The campus builds a data center or a park. $a \in \{\text{park, data center}, \cdots\}$

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:
  - an **allocation** $a \in A$
  - a **payment** vector $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. **Public decision.** The campus builds a data center or a park. $a \in \{\text{park, data center}, \cdots\}$
  2. **Divisible good.** Shares of a 5G spectrum band. $a = (a_1, \cdots, a_n)$, $a_i \in [0, 1]$, $\sum_{i \in N} a_i = 1$

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:
  - an **allocation** $a \in A$
  - a **payment** vector $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. **Public decision.** The campus builds a data center or a park. $a \in \{\text{park, data center}, \cdots\}$
  2. **Divisible good.** Shares of a 5G spectrum band. $a = (a_1, \cdots, a_n)$, $a_i \in [0, 1]$, $\sum_{i \in N} a_i = 1$
  3. **One indivisible object.** One week on a GPU pod, auctioned. $a_i \in \{0, 1\}$, $\sum_{i \in N} a_i \leqslant 1$

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$. X is the space of all **outcomes**.
- Now an outcome $x = (a, \pi) \in X$ has two parts:
  - an **allocation** $a \in A$
  - a **payment** vector $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. **Public decision.** The campus builds a data center or a park. $a \in \{\text{park, data center}, \cdots\}$
  2. **Divisible good.** Shares of a 5G spectrum band. $a = (a_1, \cdots, a_n)$, $a_i \in [0, 1]$, $\sum_{i \in N} a_i = 1$
  3. **One indivisible object.** One week on a GPU pod, auctioned. $a_i \in \{0, 1\}$, $\sum_{i \in N} a_i \leqslant 1$
  4. **Partitioning objects.** A set S of cloud VMs.
     $a = \{(S_1, \cdots, S_n) : S_i \subseteq S,\ \forall i \in N, S_i \cap S_j = \emptyset,\ \forall i \neq j\}$

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**
- Campus example, $a \in \{\text{Data center}, \text{Park}\}$:

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**
- Campus example, $a \in \{\text{Data center}, \text{Park}\}$:
  - 'environmentalist' type: $v_i(D, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**
- Campus example, $a \in \{\text{Data center}, \text{Park}\}$:
  - 'environmentalist' type: $v_i(D, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$
  - 'business' type, a startup founder: $v_i(D, \theta_i^{\text{bus}}) > v_i(P, \theta_i^{\text{bus}})$

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**
- Campus example, $a \in \{\text{Data center}, \text{Park}\}$:
  - 'environmentalist' type: $v_i(D, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$
  - 'business' type, a startup founder: $v_i(D, \theta_i^{\text{bus}}) > v_i(P, \theta_i^{\text{bus}})$
- The new instrument is **money**: payments $\pi_i \in \mathbb{R}$.

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**
- Campus example, $a \in \{\text{Data center}, \text{Park}\}$:
  - 'environmentalist' type: $v_i(D, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$
  - 'business' type, a startup founder: $v_i(D, \theta_i^{\text{bus}}) > v_i(P, \theta_i^{\text{bus}})$
- The new instrument is **money**: payments $\pi_i \in \mathbb{R}$.
  - $\pi_i > 0$: i pays. $\pi_i < 0$: i is paid. $\pi_i = 0$: nothing moves.

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**
- Campus example, $a \in \{\text{Data center}, \text{Park}\}$:
  - 'environmentalist' type: $v_i(D, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$
  - 'business' type, a startup founder: $v_i(D, \theta_i^{\text{bus}}) > v_i(P, \theta_i^{\text{bus}})$
- The new instrument is **money**: payments $\pi_i \in \mathbb{R}$.
  - $\pi_i > 0$: i pays. $\pi_i < 0$: i is paid. $\pi_i = 0$: nothing moves.
- Utility at outcome $x = (a, \pi)$:

$$u_i((a, \pi), \theta_i) = v_i(a, \theta_i) - \pi_i \qquad \textbf{(quasi-linear payoff)}$$

Types, Valuations, and Payments

- Agent i's **type** $\theta_i \in \Theta_i$ is her private information.
- Her benefit is the **valuation function** $v_i : A \times \Theta_i \to \mathbb{R}$ **(independent private values)**
- Campus example, $a \in \{\text{Data center}, \text{Park}\}$:
  - 'environmentalist' type: $v_i(D, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$
  - 'business' type, a startup founder: $v_i(D, \theta_i^{\text{bus}}) > v_i(P, \theta_i^{\text{bus}})$
- The new instrument is **money**: payments $\pi_i \in \mathbb{R}$.
  - $\pi_i > 0$: i pays. $\pi_i < 0$: i is paid. $\pi_i = 0$: nothing moves.
- Utility at outcome $x = (a, \pi)$:

$$u_i((a, \pi), \theta_i) = v_i(a, \theta_i) - \pi_i \qquad \textbf{(quasi-linear payoff)}$$

- Such types form the **quasi-linear domain**.

Quasi-Linearity Restricts the Domain

Question

Why is this a domain restriction?

Quasi-Linearity Restricts the Domain

Question

Why is this a domain restriction?

Answer

- Take (a, π) and (a, π'): same allocation, different payments.

Quasi-Linearity Restricts the Domain

Question

Why is this a domain restriction?

Answer

- Take (a, π) and (a, π'): same allocation, different payments.
- Suppose $\pi'_i < \pi_i$ for some $i \in N$.

Quasi-Linearity Restricts the Domain

Question

Why is this a domain restriction?

Answer

- Take (a, π) and (a, π'): same allocation, different payments.
- Suppose $\pi_i' < \pi_i$ for some $i \in N$.
- $v_i(a, \theta_i) - \pi_i' > v_i(a, \theta_i) - \pi_i$, for every $\theta_i \in \Theta_i$.

Quasi-Linearity Restricts the Domain

Question

Why is this a domain restriction?

Answer

- Take (a, π) and (a, π'): same allocation, different payments.
- Suppose $\pi_i' < \pi_i$ for some $i \in N$.
- $v_i(a, \theta_i) - \pi_i' > v_i(a, \theta_i) - \pi_i$, for every $\theta_i \in \Theta_i$.
- So **no** type ranks (a, π) above (a, π'). Paying less always wins.

Quasi-Linearity Restricts the Domain

Question

Why is this a domain restriction?

Answer

- Take (a, π) and (a, π'): same allocation, different payments.
- Suppose $\pi_i' < \pi_i$ for some $i \in N$.
- $v_i(a, \theta_i) - \pi_i' > v_i(a, \theta_i) - \pi_i$, for every $\theta_i \in \Theta_i$.
- So **no** type ranks (a, π) above (a, π'). Paying less always wins.
- In the unrestricted domain, both rankings were allowed.

Quasi-Linearity Restricts the Domain

Question

Why is this a domain restriction?

Answer

- Take (a, π) and (a, π'): same allocation, different payments.
- Suppose $\pi_i' < \pi_i$ for some $i \in N$.
- $v_i(a, \theta_i) - \pi_i' > v_i(a, \theta_i) - \pi_i$, for every $\theta_i \in \Theta_i$.
- So **no** type ranks (a, π) above (a, π'). Paying less always wins.
- In the unrestricted domain, both rankings were allowed.

This restriction opens the door to many non-dictatorial mechanisms.

Allocation Rules

The SCF splits as $F \equiv (f, (p_1, \ldots, p_n)) \equiv (f, p)$:

- **Allocation rule** $f : \Theta_1 \times \cdots \times \Theta_n \to A$
- **Payment rule** $p_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$, $\forall i \in N$

$$\text{Hence} \quad F : \Theta_1 \times \cdots \times \Theta_n \to A \times \mathbb{R}$$

Allocation Rules

The SCF splits as $F \equiv (f, (p_1, \ldots, p_n)) \equiv (f, p)$:

- **Allocation rule** $f : \Theta_1 \times \cdots \times \Theta_n \to A$
- **Payment rule** $p_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$, $\forall i \in N$

$$\text{Hence} \quad F : \Theta_1 \times \cdots \times \Theta_n \to A \times \mathbb{R}$$

1. **Constant:** $f^c(\theta) = a, \ \forall \theta \in \Theta$

Allocation Rules

The SCF splits as $F \equiv (f, (p_1, \ldots, p_n)) \equiv (f, p)$:

- **Allocation rule** $f : \Theta_1 \times \cdots \times \Theta_n \to A$
- **Payment rule** $p_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$, $\forall i \in N$

$$\text{Hence} \quad F : \Theta_1 \times \cdots \times \Theta_n \to A \times \mathbb{R}$$

1. **Constant:** $f^c(\theta) = a$, $\forall \theta \in \Theta$
2. **Dictatorial:** $f^D(\theta) \in \arg\max_{a \in A} v_d(a, \theta_d)$, for some $d \in N$

Allocation Rules

The SCF splits as $F \equiv (f, (p_1, \ldots, p_n)) \equiv (f, p)$:

- **Allocation rule** $f : \Theta_1 \times \cdots \times \Theta_n \to A$
- **Payment rule** $p_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$, $\forall i \in N$

$$\text{Hence} \quad F : \Theta_1 \times \cdots \times \Theta_n \to A \times \mathbb{R}$$

1. **Constant:** $f^c(\theta) = a, \ \forall \theta \in \Theta$
2. **Dictatorial:** $f^D(\theta) \in \arg\max_{a \in A} v_d(a, \theta_d)$, for some $d \in N$
3. **Allocatively efficient (utilitarian):** $f^{AE}(\theta) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, \theta_i)$
   Note: not the same as PE. PE is about outcomes, so it also sees payments.

Allocation Rules

The SCF splits as $F \equiv (f, (p_1, \ldots, p_n)) \equiv (f, p)$:

- **Allocation rule** $f : \Theta_1 \times \cdots \times \Theta_n \to A$
- **Payment rule** $p_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$, $\forall i \in N$

$$\text{Hence} \quad F : \Theta_1 \times \cdots \times \Theta_n \to A \times \mathbb{R}$$

1. **Constant:** $f^c(\theta) = a, \ \forall \theta \in \Theta$
2. **Dictatorial:** $f^D(\theta) \in \arg\max_{a \in A} v_d(a, \theta_d)$, for some $d \in N$
3. **Allocatively efficient (utilitarian):** $f^{AE}(\theta) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, \theta_i)$
   Note: not the same as PE. PE is about outcomes, so it also sees payments.
4. **Affine maximizer:** $f^{AM}(\theta) \in \arg\max_{a \in A} \left(\sum_{i \in N} \lambda_i v_i(a, \theta_i) + \kappa(a) \right)$, $\lambda_i \geqslant 0$, not all zero
   $\lambda_i \equiv 1, \kappa \equiv 0$: AE. $\lambda_d = 1$, rest 0, $\kappa \equiv 0$: dictatorial.

Allocation Rules

The SCF splits as $F \equiv (f, (p_1, \ldots, p_n)) \equiv (f, p)$:

- **Allocation rule** $f : \Theta_1 \times \cdots \times \Theta_n \to A$
- **Payment rule** $p_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$, $\forall i \in N$

$$\text{Hence} \quad F : \Theta_1 \times \cdots \times \Theta_n \to A \times \mathbb{R}$$

1. **Constant:** $f^c(\theta) = a, \ \forall \theta \in \Theta$
2. **Dictatorial:** $f^D(\theta) \in \arg\max_{a \in A} v_d(a, \theta_d)$, for some $d \in N$
3. **Allocatively efficient (utilitarian):** $f^{AE}(\theta) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, \theta_i)$
   Note: not the same as PE. PE is about outcomes, so it also sees payments.
4. **Affine maximizer:** $f^{AM}(\theta) \in \arg\max_{a \in A} \left(\sum_{i \in N} \lambda_i v_i(a, \theta_i) + \kappa(a) \right)$, $\lambda_i \geqslant 0$, not all zero
   $\lambda_i \equiv 1, \kappa \equiv 0$: AE. $\lambda_d = 1$, rest 0, $\kappa \equiv 0$: dictatorial.
5. **Max-min (egalitarian):** $f^{MM}(\theta) \in \arg\max_{a \in A} \min_{i \in N} v_i(a, \theta_i)$

Payment Rules

1. **No deficit**: $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
   The platform never loses money.

Payment Rules

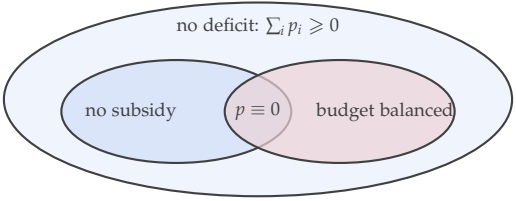
1. **No deficit**: $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
   The platform never loses money.
2. **No subsidy**: $p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta, \ \forall i \in N$
   Nobody is ever paid.

Payment Rules

1. **No deficit**: $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
   The platform never loses money.
2. **No subsidy**: $p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta, \ \forall i \in N$
   Nobody is ever paid.
3. **Budget balanced**: $\sum_{i \in N} p_i(\theta) = 0, \ \forall \theta \in \Theta$
   Money only moves between agents.

Payment Rules

1. **No deficit**: $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
   The platform never loses money.
2. **No subsidy**: $p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta, \ \forall i \in N$
   Nobody is ever paid.
3. **Budget balanced**: $\sum_{i \in N} p_i(\theta) = 0, \ \forall \theta \in \Theta$
   Money only moves between agents.



both no subsidy and budget balance imply no deficit



Chunk 4

DSIC, Efficiency, and Groves Payments

(≈ 25 minutes)

DSIC

Definition (DSIC)

A mechanism (f, p) is **dominant strategy incentive compatible (DSIC)** if

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}),$$

$\forall \tilde{\theta}_{-i} \in \Theta_{-i},\ \theta_i', \theta_i \in \Theta_i,\ \forall i \in N.$

- Truth-telling is a weakly dominant strategy equilibrium.

DSIC

Definition (DSIC)

A mechanism (f, p) is **dominant strategy incentive compatible (DSIC)** if

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}),$$

$\forall \tilde{\theta}_{-i} \in \Theta_{-i},\ \theta_i', \theta_i \in \Theta_i,\ \forall i \in N.$

- Truth-telling is a weakly dominant strategy equilibrium.
- We say p **implements** f in dominant strategies.

DSIC

Definition (DSIC)

A mechanism (f, p) is **dominant strategy incentive compatible (DSIC)** if

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}),$$

$\forall \tilde{\theta}_{-i} \in \Theta_{-i},\ \theta_i', \theta_i \in \Theta_i,\ \forall i \in N.$

- Truth-telling is a weakly dominant strategy equilibrium.
- We say p **implements** f in dominant strategies.
- In the QL domain, we mostly care about f. Then we hunt for a p.

DSIC

Definition (DSIC)

A mechanism (f, p) is **dominant strategy incentive compatible (DSIC)** if

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}),$$

$\forall \tilde{\theta}_{-i} \in \Theta_{-i},\ \theta_i', \theta_i \in \Theta_i,\ \forall i \in N.$

- Truth-telling is a weakly dominant strategy equilibrium.
- We say p **implements** f in dominant strategies.
- In the QL domain, we mostly care about f. Then we hunt for a p.

Example. $N = \{1, 2\}$, $\Theta_1 = \Theta_2 = \{\theta^H, \theta^L\}$. For player 1, $\forall \theta_2 \in \Theta_2$:

$$v_1(f(\theta^H, \theta_2), \theta^H) - p_1(\theta^H, \theta_2) \geqslant v_1(f(\theta^L, \theta_2), \theta^H) - p_1(\theta^L, \theta_2)$$
$$v_1(f(\theta^L, \theta_2), \theta^L) - p_1(\theta^L, \theta_2) \geqslant v_1(f(\theta^H, \theta_2), \theta^L) - p_1(\theta^H, \theta_2)$$

Two more for player 2. Every true type beats every lie, against every report of the other.

Properties of the Payment

- Say (f, p) is DSIC. Consider another payment:

$$q_i(\theta_i, \theta_{-i}) = p_i(\theta_i, \theta_{-i}) + h_i(\theta_{-i}), \quad \forall \theta, \forall i \in N$$

Properties of the Payment

- Say (f, p) is DSIC. Consider another payment:

$$q_i(\theta_i, \theta_{-i}) = p_i(\theta_i, \theta_{-i}) + h_i(\theta_{-i}), \quad \forall \theta, \forall i \in N$$

- Is (f, q) DSIC? Yes. $h_i(\tilde{\theta}_{-i})$ appears on both sides and cancels:

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i})$$

Properties of the Payment

- Say (f, p) is DSIC. Consider another payment:

$$q_i(\theta_i, \theta_{-i}) = p_i(\theta_i, \theta_{-i}) + h_i(\theta_{-i}), \quad \forall \theta, \forall i \in N$$

- Is (f, q) DSIC? Yes. $h_i(\tilde{\theta}_{-i})$ appears on both sides and cancels:

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i})$$

- One implementing payment gives uncountably many.

Properties of the Payment

- Say (f, p) is DSIC. Consider another payment:

$$q_i(\theta_i, \theta_{-i}) = p_i(\theta_i, \theta_{-i}) + h_i(\theta_{-i}), \quad \forall \theta, \forall i \in N$$

- Is (f, q) DSIC? Yes. $h_i(\tilde{\theta}_{-i})$ appears on both sides and cancels:

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i})$$

- One implementing payment gives uncountably many.
- **The converse question:** when do all payments that implement f differ only by some $h_i(\theta_{-i})$?

Properties of the Payment

- Say (f, p) is DSIC. Consider another payment:

$$q_i(\theta_i, \theta_{-i}) = p_i(\theta_i, \theta_{-i}) + h_i(\theta_{-i}), \quad \forall \theta, \forall i \in N$$

- Is (f, q) DSIC? Yes. $h_i(\tilde{\theta}_{-i})$ appears on both sides and cancels:

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}) - h_i(\tilde{\theta}_{-i})$$

- One implementing payment gives uncountably many.
- **The converse question:** when do all payments that implement f differ only by some $h_i(\theta_{-i})$?
- **A fact [exercise]**: if $f(\theta) = f(\tilde{\theta}_i, \theta_{-i}) = a$ and p implements f, then $p_i(\theta) = p_i(\tilde{\theta}_i, \theta_{-i})$.

Pareto Optimality in the Quasi-Linear Domain



Definition (Pareto Optimal)

$(f, (p_1, \ldots, p_n))$ is **Pareto optimal** if for every $\theta \in \Theta$, there is no allocation $b \neq f(\theta)$ and payments $(\pi_1, \ldots, \pi_n)$ such that

$$v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta), \ \forall i \in N,$$

with the inequality strict for some $i \in N$.

Pareto Optimality in the Quasi-Linear Domain

Definition (Pareto Optimal)

$(f, (p_1, \ldots, p_n))$ is **Pareto optimal** if for every $\theta \in \Theta$, there is no allocation $b \neq f(\theta)$ and payments $(\pi_1, \ldots, \pi_n)$ with $\sum_{i \in N} \pi_i \geqslant \sum_{i \in N} p_i(\theta)$ such that

$$v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta), \ \forall i \in N,$$

with the inequality strict for some $i \in N$.

- Why the red condition? Without it, just pay everyone a lot. Everyone is happier.
- So we ask: can we do better *without spending more*?

Pareto Optimality in the Quasi-Linear Domain

Definition (Pareto Optimal)

$(f, (p_1, \ldots, p_n))$ is **Pareto optimal** if for every $\theta \in \Theta$, there is no allocation $b \neq f(\theta)$ and payments $(\pi_1, \ldots, \pi_n)$ with $\sum_{i \in N} \pi_i \geqslant \sum_{i \in N} p_i(\theta)$ such that

$$v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta), \ \forall i \in N,$$

with the inequality strict for some $i \in N$.

- Why the red condition? Without it, just pay everyone a lot. Everyone is happier.
- So we ask: can we do better *without spending more*?

Theorem

*A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.*

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show $\neg$PO $\implies$ $\neg$AE.

- $\neg$PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show $\neg$PO $\implies$ $\neg$AE.

- $\neg$PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\begin{aligned} \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) &> \sum_i \pi_i - \sum_i p_i(\theta) \\ &\geqslant 0 \end{aligned}$$

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show $\neg$PO $\implies$ $\neg$AE.

- $\neg$PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\begin{aligned} \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) &> \sum_i \pi_i - \sum_i p_i(\theta) \\ &\geqslant 0 \end{aligned}$$

- So f is not AE.

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show $\neg$PO $\implies$ $\neg$AE.

- $\neg$PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > \sum_i \pi_i - \sum_i p_i(\theta) \geqslant 0$$

- So f is not AE.

($\Longrightarrow$) Show $\neg$AE $\implies$ $\neg$PO.

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show $\neg$PO $\implies$ $\neg$AE.

- $\neg$PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > \sum_i \pi_i - \sum_i p_i(\theta) \geqslant 0$$

- So f is not AE.

($\Longrightarrow$) Show $\neg$AE $\implies$ $\neg$PO.

- $\neg$AE: $\exists\, \theta, b$ with $\delta = \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > 0$.

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show $\neg$PO $\implies$ $\neg$AE.

- $\neg$PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\begin{aligned} \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) &> \sum_i \pi_i - \sum_i p_i(\theta) \\ &\geqslant 0 \end{aligned}$$

- So f is not AE.

($\Longrightarrow$) Show $\neg$AE $\implies$ $\neg$PO.

- $\neg$AE: $\exists\, \theta, b$ with $\delta = \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > 0$.
- Set $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n$.

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show ¬PO $\implies$ ¬AE.

- ¬PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > \sum_i \pi_i - \sum_i p_i(\theta) \geqslant 0$$

- So f is not AE.

($\Longrightarrow$) Show ¬AE $\implies$ ¬PO.

- ¬AE: $\exists\, \theta, b$ with $\delta = \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > 0$.
- Set $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n$.
- Then everyone gains exactly $\delta/n > 0$.

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show ¬PO $\implies$ ¬AE.

- ¬PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > \sum_i \pi_i - \sum_i p_i(\theta) \geqslant 0$$

- So f is not AE.

($\Longrightarrow$) Show ¬AE $\implies$ ¬PO.

- ¬AE: $\exists\, \theta, b$ with $\delta = \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > 0$.
- Set $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n$.
- Then everyone gains exactly $\delta/n > 0$.
- And $\sum_i \pi_i = \sum_i p_i(\theta)$: same budget.

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show ¬PO $\implies$ ¬AE.

- ¬PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > \sum_i \pi_i - \sum_i p_i(\theta) \geqslant 0$$

- So f is not AE.

($\Longrightarrow$) Show ¬AE $\implies$ ¬PO.

- ¬AE: $\exists\, \theta, b$ with $\delta = \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > 0$.
- Set $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n$.
- Then everyone gains exactly $\delta/n > 0$.
- And $\sum_i \pi_i = \sum_i p_i(\theta)$: same budget.
- So f is not PO. □

Proof: PO $\iff$ AE

A mechanism $(f, (p_1, \cdots, p_n))$ is **Pareto optimal** iff it is allocatively efficient.

($\Longleftarrow$) Show ¬PO $\implies$ ¬AE.

- ¬PO: $\exists\, b, \pi, \theta$ with $\sum_i \pi_i \geqslant \sum_i p_i(\theta)$ and $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta)$, $\forall i$, strict for some j.
- Sum over i:

$$\sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > \sum_i \pi_i - \sum_i p_i(\theta) \geqslant 0$$

- So f is not AE.

($\Longrightarrow$) Show ¬AE $\implies$ ¬PO.

- ¬AE: $\exists\, \theta, b$ with $\delta = \sum_i v_i(b, \theta_i) - \sum_i v_i(f(\theta), \theta_i) > 0$.
- Set $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n$.
- Then everyone gains exactly $\delta/n > 0$.
- And $\sum_i \pi_i = \sum_i p_i(\theta)$: same budget.
- So f is not PO. □

Takeaway

Split the extra welfare δ equally. Money makes efficiency a question about f alone.

Groves Payments

Can we implement f^{AE} truthfully? Try:

$$p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j), \quad h_i : \Theta_{-i} \to \mathbb{R} \text{ arbitrary.}$$

This is the **Groves payment**. You are paid the welfare of everyone else.

Groves Payments

Can we implement f^{AE} truthfully? Try:

$$p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j), \quad h_i : \Theta_{-i} \to \mathbb{R} \text{ arbitrary.}$$

This is the **Groves payment**. You are paid the welfare of everyone else.

Example (Four AI labs, one GPU pod for a week)

- $N = \{1, 2, 3, 4\}$. Values: $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$ if they get the pod, 0 otherwise.

Groves Payments

Can we implement f^{AE} truthfully? Try:

$$p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j), \quad h_i : \Theta_{-i} \to \mathbb{R} \text{ arbitrary.}$$

This is the **Groves payment**. You are paid the welfare of everyone else.

Example (Four AI labs, one GPU pod for a week)

- $N = \{1, 2, 3, 4\}$. Values: $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$ if they get the pod, 0 otherwise.
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$: $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$.

Groves Payments

Can we implement f^{AE} truthfully? Try:

$$p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j), \quad h_i : \Theta_{-i} \to \mathbb{R} \text{ arbitrary.}$$

This is the **Groves payment**. You are paid the welfare of everyone else.

Example (Four AI labs, one GPU pod for a week)

- $N = \{1, 2, 3, 4\}$. Values: $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$ if they get the pod, 0 otherwise.
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$: $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$.
- Efficient allocation: lab 1 gets the pod.

Groves Payments

Can we implement f^{AE} truthfully? Try:

$$p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j), \quad h_i : \Theta_{-i} \to \mathbb{R} \text{ arbitrary.}$$

This is the **Groves payment**. You are paid the welfare of everyone else.

Example (Four AI labs, one GPU pod for a week)

- $N = \{1, 2, 3, 4\}$. Values: $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$ if they get the pod, 0 otherwise.
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$: $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$.
- Efficient allocation: lab 1 gets the pod.
- $p_1 = 4 - 0 = 4$, $p_2 = 4 - 10 = -6$, $p_3 = 4 - 10 = -6$, $p_4 = 6 - 10 = -4$.

Groves Payments

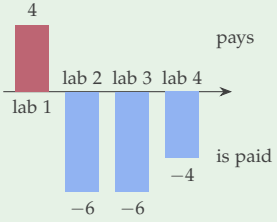
Can we implement f^{AE} truthfully? Try:

$$p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j), \quad h_i : \Theta_{-i} \to \mathbb{R} \text{ arbitrary.}$$

This is the **Groves payment**. You are paid the welfare of everyone else.

Example (Four AI labs, one GPU pod for a week)

- $N = \{1, 2, 3, 4\}$. Values: $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$ if they get the pod, 0 otherwise.
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$: $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$.
- Efficient allocation: lab 1 gets the pod.
- $p_1 = 4 - 0 = 4$, $p_2 = 4 - 10 = -6$, $p_3 = 4 - 10 = -6$, $p_4 = 6 - 10 = -4$.



Groves Payments

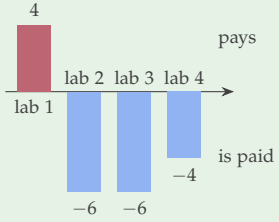
Can we implement f^{AE} truthfully? Try:

$$p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j), \quad h_i : \Theta_{-i} \to \mathbb{R} \text{ arbitrary.}$$

This is the **Groves payment**. You are paid the welfare of everyone else.

Example (Four AI labs, one GPU pod for a week)

- $N = \{1, 2, 3, 4\}$. Values: $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$ if they get the pod, 0 otherwise.
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$: $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$.
- Efficient allocation: lab 1 gets the pod.
- $p_1 = 4 - 0 = 4$, $p_2 = 4 - 10 = -6$, $p_3 = 4 - 10 = -6$, $p_4 = 6 - 10 = -4$.



Only the winner pays. The others get paid. **Surprisingly, this is truthful.**

Groves Mechanisms Are Truthful



Theorem

Groves mechanisms are DSIC.

- Fix player i. Let $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$ and $f^{AE}(\theta_i', \tilde{\theta}_{-i}) = b$.

Groves Mechanisms Are Truthful

Theorem

Groves mechanisms are DSIC.

- Fix player i. Let $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$ and $f^{AE}(\theta_i', \tilde{\theta}_{-i}) = b$.
- By efficiency at the true type, $v_i(a, \theta_i) + \sum_{j \neq i} v_j(a, \tilde{\theta}_j) \geqslant v_i(b, \theta_i) + \sum_{j \neq i} v_j(b, \tilde{\theta}_j)$.

Groves Mechanisms Are Truthful

Theorem

Groves mechanisms are DSIC.

- Fix player i. Let $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$ and $f^{AE}(\theta_i', \tilde{\theta}_{-i}) = b$.
- By efficiency at the true type, $v_i(a, \theta_i) + \sum_{j \neq i} v_j(a, \tilde{\theta}_j) \geqslant v_i(b, \theta_i) + \sum_{j \neq i} v_j(b, \tilde{\theta}_j)$.
- Utility of i when she reports θ_i:

$$
\begin{aligned}
& v_i(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \\
&= v_i(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \theta_i) - h_i(\tilde{\theta}_{-i}) + \textstyle\sum_{j \neq i} v_j(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \tilde{\theta}_j) \\
&\geqslant v_i(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \theta_i) - h_i(\tilde{\theta}_{-i}) + \textstyle\sum_{j \neq i} v_j(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \tilde{\theta}_j) \\
&= v_i(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i})
\end{aligned}
$$

Groves Mechanisms Are Truthful

Theorem

Groves mechanisms are DSIC.

- Fix player i. Let $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$ and $f^{AE}(\theta_i', \tilde{\theta}_{-i}) = b$.
- By efficiency at the true type, $v_i(a, \theta_i) + \sum_{j \neq i} v_j(a, \tilde{\theta}_j) \geqslant v_i(b, \theta_i) + \sum_{j \neq i} v_j(b, \tilde{\theta}_j)$.
- Utility of i when she reports θ_i:

$$\begin{aligned}
& v_i(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \\
&= v_i(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \theta_i) - h_i(\tilde{\theta}_{-i}) + \textstyle\sum_{j \neq i} v_j(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \tilde{\theta}_j) \\
&\geqslant v_i(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \theta_i) - h_i(\tilde{\theta}_{-i}) + \textstyle\sum_{j \neq i} v_j(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \tilde{\theta}_j) \\
&= v_i(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i})
\end{aligned}$$

- Player i was arbitrary. Hence the claim. □

Groves Mechanisms Are Truthful

Theorem

Groves mechanisms are DSIC.

- Fix player i. Let $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$ and $f^{AE}(\theta_i', \tilde{\theta}_{-i}) = b$.
- By efficiency at the true type, $v_i(a, \theta_i) + \sum_{j \neq i} v_j(a, \tilde{\theta}_j) \geqslant v_i(b, \theta_i) + \sum_{j \neq i} v_j(b, \tilde{\theta}_j)$.
- Utility of i when she reports θ_i:

$$\begin{aligned}
& v_i(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \\
&= v_i(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \theta_i) - h_i(\tilde{\theta}_{-i}) + \textstyle\sum_{j \neq i} v_j(f^{AE}(\theta_i, \tilde{\theta}_{-i}), \tilde{\theta}_j) \\
&\geqslant v_i(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \theta_i) - h_i(\tilde{\theta}_{-i}) + \textstyle\sum_{j \neq i} v_j(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \tilde{\theta}_j) \\
&= v_i(f^{AE}(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i})
\end{aligned}$$

- Player i was arbitrary. Hence the claim. □

Why it works: i's utility is total welfare minus $h_i(\tilde{\theta}_{-i})$, a term she cannot touch. Her goal is f^{AE}'s goal.

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 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.

Lecture 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.
- Task sharing: serial dictatorship is not ANON. Proportional is not SP.

Lecture 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.
- Task sharing: serial dictatorship is not ANON. Proportional is not SP.
- The uniform rule is the unique SP, PE, ANON rule. It is also EF and fast.

Lecture 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.
- Task sharing: serial dictatorship is not ANON. Proportional is not SP.
- The uniform rule is the unique SP, PE, ANON rule. It is also EF and fast.
- With money: outcome = (allocation, payment). Quasi-linear utility $v_i - \pi_i$.

Lecture 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.
- Task sharing: serial dictatorship is not ANON. Proportional is not SP.
- The uniform rule is the unique SP, PE, ANON rule. It is also EF and fast.
- With money: outcome = (allocation, payment). Quasi-linear utility $v_i - \pi_i$.
- Adding $h_i(\theta_{-i})$ to a DSIC payment keeps it DSIC.

Lecture 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.
- Task sharing: serial dictatorship is not ANON. Proportional is not SP.
- The uniform rule is the unique SP, PE, ANON rule. It is also EF and fast.
- With money: outcome = (allocation, payment). Quasi-linear utility $v_i - \pi_i$.
- Adding $h_i(\theta_{-i})$ to a DSIC payment keeps it DSIC.
- In the QL domain, PO $\iff$ AE.

Lecture 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.
- Task sharing: serial dictatorship is not ANON. Proportional is not SP.
- The uniform rule is the unique SP, PE, ANON rule. It is also EF and fast.
- With money: outcome = (allocation, payment). Quasi-linear utility $v_i - \pi_i$.
- Adding $h_i(\theta_{-i})$ to a DSIC payment keeps it DSIC.
- In the QL domain, PO $\iff$ AE.
- Groves payments implement f^{AE} in dominant strategies.

Lecture 18 and Week 9: What We Built

- Moulin: SP + ONTO + ANON on single-peaked domains = median voter rules.
- Task sharing: serial dictatorship is not ANON. Proportional is not SP.
- The uniform rule is the unique SP, PE, ANON rule. It is also EF and fast.
- With money: outcome = (allocation, payment). Quasi-linear utility $v_i - \pi_i$.
- Adding $h_i(\theta_{-i})$ to a DSIC payment keeps it DSIC.
- In the QL domain, PO $\iff$ AE.
- Groves payments implement f^{AE} in dominant strategies.
- **Next week:** the VCG mechanism, a special Groves mechanism.



End of Lecture 18

See you next class.



भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay