भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay

CS 6001: Game Theory and Algorithmic Mechanism Design

Week 9

Swaprava Nath

Slide preparation acknowledgments: Rounak Dalmia

ज्ञानम् परमम् ध्येयम्

Knowledge is the supreme goal

Contents

Task Allocation Domain

- Unit amount of task to be shared among n agents

Task Allocation Domain

- Unit amount of task to be shared among n agents
- Agent i gets a share $s_i \in [0, 1]$ of the job, $\sum_{i \in N} s_i = 1$

Task Allocation Domain

- Unit amount of task to be shared among n agents
- Agent i gets a share $s_i \in [0, 1]$ of the job, $\sum_{i \in N} s_i = 1$
- Agent **payoff**: every agent has a most preferred share of work.

Task Allocation Domain

- Unit amount of task to be shared among n agents
- Agent i gets a share $s_i \in [0,1]$ of the job, $\sum_{i \in N} s_i = 1$
- Agent **payoff**: every agent has a most preferred share of work.
- **Example:**

Task Allocation Domain

- Unit amount of task to be shared among n agents
- Agent i gets a share $s_i \in [0, 1]$ of the job, $\sum_{i \in N} s_i = 1$
- Agent **payoff**: every agent has a most preferred share of work.
- **Example:**
  - The task has rewards, e.g., wages per unit time $= w$

Task Allocation Domain

- Unit amount of task to be shared among n agents
- Agent i gets a share $s_i \in [0,1]$ of the job, $\sum_{i \in N} s_i = 1$
- Agent **payoff**: every agent has a most preferred share of work.
- **Example:**
  - The task has rewards, e.g., wages per unit time $= w$
  - if agent i works for t_i time then gets $w \cdot t_i$

Task Allocation Domain

- Unit amount of task to be shared among n agents
- Agent i gets a share $s_i \in [0, 1]$ of the job, $\sum_{i \in N} s_i = 1$
- Agent **payoff**: every agent has a most preferred share of work.
- **Example:**
  - The task has rewards, e.g., wages per unit time $= w$
  - if agent i works for t_i time then gets $w \cdot t_i$
  - The task also has costs, e.g., physical tiredness/less free time, etc. Let the cost be quadratic = $c_i t_i^2$

Task Allocation Domain

- Unit amount of task to be shared among n agents
- Agent i gets a share $s_i \in [0, 1]$ of the job, $\sum_{i \in N} s_i = 1$
- Agent **payoff**: every agent has a most preferred share of work.
- **Example:**
  - The task has rewards, e.g., wages per unit time $= w$
  - if agent i works for t_i time then gets $w \cdot t_i$
  - The task also has costs, e.g., physical tiredness/less free time, etc. Let the cost be quadratic = $c_i t_i^2$
  - Net payoff = $wt_i - c_i t_i^2 \implies$ **maximized** at $t_i = w/2c_i$, and **monotone** decreasing on both sides

Task Allocation Domain

- Net payoff = wt_i - $c_i t_i^2 \implies$ **maximized** at $t_i = w/2c_i$

Task Allocation Domain

- Net payoff = wt_i - $c_i t_i^2 \implies$ **maximized** at $t_i = w/2c_i$
- **Important:** This is single peaked over the **share of the task** and not over the alternatives

Task Allocation Domain

- Net payoff = wt_i - $c_i t_i^2 \implies$ **maximized** at $t_i = w/2c_i$
- **Important:** This is single peaked over the **share of the task** and not over the alternatives
- Suppose, two alternatives are $(0.2, 0.4, 0.4)$ and $(0.2, 0.6, 0.2)$: player 1 likes both of them equally

Task Allocation Domain

- Net payoff = wt_i - $c_i t_i^2 \implies$ **maximized** at $t_i = w/2c_i$
- **Important:** This is single peaked over the **share of the task** and not over the alternatives
- Suppose, two alternatives are $(0.2, 0.4, 0.4)$ and $(0.2, 0.6, 0.2)$: player 1 likes both of them equally
- For 3 players, the set of alternatives is a simplex

Task Allocation Domain

- Net payoff = wt_i - $c_i t_i^2 \implies$ **maximized** at $t_i = w/2c_i$
- **Important:** This is single peaked over the **share of the task** and not over the alternatives
- Suppose, two alternatives are $(0.2, 0.4, 0.4)$ and $(0.2, 0.6, 0.2)$: player 1 likes both of them equally
- For 3 players, the set of alternatives is a simplex
- There cannot be a single common order over the alternatives s.t. the preferences are single-peaked for all agents

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$
- Let $P \in T^n$

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$
- Let $P \in T^n$
  - $f(P) = (f_1(P), f_2(P), \ldots, f_n(P))$

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$
- Let $P \in T^n$
  - $f(P) = (f_1(P), f_2(P), \dots, f_n(P))$
  - $f_i(P) \in [0,1], \ \forall i \in N$

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$
- Let $P \in T^n$
  - $f(P) = (f_1(P), f_2(P), \ldots, f_n(P))$
  - $f_i(P) \in [0,1], \ \forall i \in N$
  - $\sum_{i \in N} f_i(P) = 1$

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$
- Let $P \in T^n$
  - $f(P) = (f_1(P), f_2(P), \dots, f_n(P))$
  - $f_i(P) \in [0,1],\ \forall i \in N$
  - $\sum_{i \in N} f_i(P) = 1$
- Player i has a peak p_i over the shares of the task

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0, 1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$
- Let $P \in T^n$
  - $f(P) = (f_1(P), f_2(P), \dots, f_n(P))$
  - $f_i(P) \in [0, 1], \ \forall i \in N$
  - $\sum_{i \in N} f_i(P) = 1$
- Player i has a peak p_i over the shares of the task

Task Allocation Domain and Pareto Efficiency

- Denote this **domain of task allocation** with T
- An allocation of the task is $a = (a_i \in [0,1], i \in N)$, set of all task allocations is A
- **SCF:** $f : T^n \to A$
- Let $P \in T^n$
  - $f(P) = (f_1(P), f_2(P), \dots, f_n(P))$
  - $f_i(P) \in [0,1], \ \forall i \in N$
  - $\sum_{i \in N} f_i(P) = 1$
- Player i has a peak p_i over the shares of the task

Definition (Pareto Efficiency)

An SCF f is *Pareto efficient* (PE) if there does not exist any profile P where there exists a task allocation $a \in A$ such that it is weakly preferred over $f(P)$ by all agents and strictly preferred by at least one. Mathematically,

$$\nexists P, \text{ where } \exists a \in A \text{ s.t. } \begin{array}{ll} a \, R_i \, f(P) & \forall i \in N, \\ a \, P_j \, f(P) & \exists j \in N. \end{array}$$

Implications of Pareto Efficiency

1. If $\sum_{i \in N} p_i = 1$, allocate tasks according to the peaks of the agents This is the unique PE allocation

Implications of Pareto Efficiency

1. If $\sum_{i \in N} p_i = 1$, allocate tasks according to the peaks of the agents This is the unique PE allocation
2. If $\sum_{i \in N} p_i > 1$, there must exist $k \in N$, s.t. $f_k(P) < p_k$

Implications of Pareto Efficiency

1. If $\sum_{i \in N} p_i = 1$, allocate tasks according to the peaks of the agents This is the unique PE allocation
2. If $\sum_{i \in N} p_i > 1$, there must exist $k \in N$, s.t. $f_k(P) < p_k$

Question

Can there be an agent j s.t. $f_j(P) > p_j$ if f is PE?

Implications of Pareto Efficiency

1. If $\sum_{i \in N} p_i = 1$, allocate tasks according to the peaks of the agents This is the unique PE allocation
2. If $\sum_{i \in N} p_i > 1$, there must exist $k \in N$, s.t. $f_k(P) < p_k$

Question

Can there be an agent j s.t. $f_j(P) > p_j$ if f is PE?

Answer

No. If such a j exists, increasing k's share of task and reducing j's makes both players strictly better off
Therefore, $\forall j \in N, f_j(P) \leqslant p_j$

3. If $\sum_{i \in N} p_i < 1$, by a similar argument, we conclude that $\forall j \in N, f_j(P) \geqslant p_j$

Task Allocation Domain and Anonymity

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.,

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

Task Allocation Domain and Anonymity

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:

- $N = \{1, 2, 3\},\ \sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1$
- $P = (0.7, 0.4, 0.3) \implies P^{\sigma} = (0.3, 0.7, 0.4)$

Task Allocation Domain and Anonymity

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:

- $N = \{1,2,3\},\ \sigma(1) = 2, \sigma(2) = 3, \sigma(3) = 1$
- $P = (0.7, 0.4, 0.3) \implies P^{\sigma} = (0.3, 0.7, 0.4)$
- $f_1(0.7, 0.4, 0.3) = f_2(0.3, 0.7, 0.4)$
- $f_2(0.7, 0.4, 0.3) = f_3(0.3, 0.7, 0.4)$
- $f_3(0.7, 0.4, 0.3) = f_1(0.3, 0.7, 0.4)$

Strategyproofness

Manipulability: an SCF f is **manipulable** if $\exists i \in N$ and a profile P such that, $f(P_i', P_{-i})\ P_i\ f(P_i, P_{-i})$, for some P_i'.

Strategyproofness

Manipulability: an SCF f is **manipulable** if $\exists i \in N$ and a profile P such that, $f(P'_i, P_{-i}) \; P_i \; f(P_i, P_{-i})$, for some P'_i.

Definition (Strategyproof)

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

Strategyproofness

Manipulability: an SCF f is **manipulable** if $\exists i \in N$ and a profile P such that, $f(P'_i, P_{-i}) \; P_i \; f(P_i, P_{-i})$, for some P'_i.

Definition (Strategyproof)

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

Strategyproofness (equivalent definition):

$$f(P_i, P_{-i}) \; P_i \; f(P'_i, P_{-i}) \quad \text{OR} \quad f_i(P_i, P_{-i}) = f_i(P'_i, P_{-i}), \forall P_i, P'_i \in T, \forall i \in N, \forall P_{-i} \in T^{n-1}.$$

Task Allocation Domain: Some Candidate SCFs

Definition (Serial Dictatorship)

A predetermined sequence of the agents is fixed. Each agent is given either her peak share or the leftover share of the task. If $\sum_{i \in N} p_i < 1$, then the last agent is given the leftover share.

Task Allocation Domain: Some Candidate SCFs

Definition (Serial Dictatorship)

A predetermined sequence of the agents is fixed. Each agent is given either her peak share or the leftover share of the task. If $\sum_{i \in N} p_i < 1$, then the last agent is given the leftover share.

Question

PE, SP, ANON?

Task Allocation Domain: Some Candidate SCFs

Definition (Serial Dictatorship)

A predetermined sequence of the agents is fixed. Each agent is given either her peak share or the leftover share of the task. If $\sum_{i \in N} p_i < 1$, then the last agent is given the leftover share.

Question

PE, SP, ANON?

Answer

Not ANON. Also quite unfair to the last agent.

Task Allocation Domain: Some Candidate SCFs

Definition (Proportional)

Every player is assigned a share that is c times their peaks, s.t. $c \sum_{i \in N} p_i = 1$

Task Allocation Domain: Some Candidate SCFs

Definition (Proportional)

Every player is assigned a share that is c times their peaks, s.t. $c \sum_{i \in N} p_i = 1$

Question

PE, ANON, SP?

Task Allocation Domain: Some Candidate SCFs

Definition (Proportional)

Every player is assigned a share that is c times their peaks, s.t. $c \sum_{i \in N} p_i = 1$

Question

PE, ANON, SP?

Answer

Not SP.
Suppose peaks are $0.2, 0.3, 0.1$ for 3 players, $c = 1/0.6$

Task Allocation Domain: Some Candidate SCFs

Definition (Proportional)

Every player is assigned a share that is c times their peaks, s.t. $c \sum_{i \in N} p_i = 1$

Question

PE, ANON, SP?

Answer

Not SP.
Suppose peaks are $0.2, 0.3, 0.1$ for 3 players, $c = 1/0.6$
Player 1 gets $1/3$ (more than its peak 0.2)

Task Allocation Domain: Some Candidate SCFs

Definition (Proportional)

Every player is assigned a share that is c times their peaks, s.t. $c \sum_{i \in N} p_i = 1$

Question

PE, ANON, SP?

Answer

Not SP.
Suppose peaks are $0.2, 0.3, 0.1$ for 3 players, $c = 1/0.6$
Player 1 gets $1/3$ (more than its peak 0.2)
if the report is $0.1, 0.3, 0.1$, $c = 1/0.5$, player 1 gets 0.2

Contents

PE, ANON, and SP?

Question

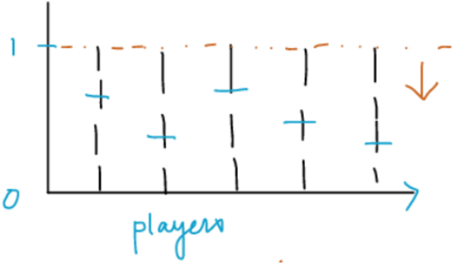
How to ensure PE, ANON, and SP in the task allocation domain?

PE, ANON, and SP?

Question

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

- Suppose, $\sum_{i \in N} p_i < 1$



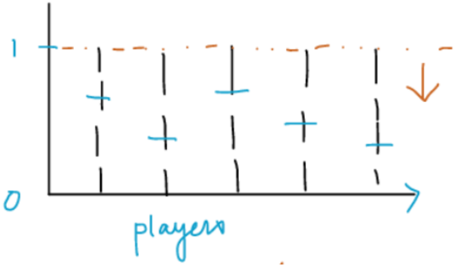
Uniform Rule (Sprumont 1991)

PE, ANON, and SP?

Question

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

- Suppose, $\sum_{i \in N} p_i < 1$
- Begin with everyone's allocation being 1 (infeasible), i.e., $f_i(P) = 1, \forall i \in N$



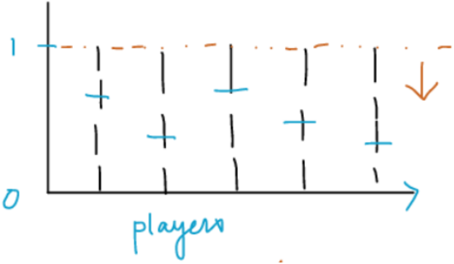
Uniform Rule (Sprumont 1991)

PE, ANON, and SP?

Question

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

- Suppose, $\sum_{i \in N} p_i < 1$
- Begin with everyone's allocation being 1 (infeasible), i.e., $f_i(P) = 1, \forall i \in N$
- Keep reducing until $\sum_{i \in N} f_i(P) = 1$



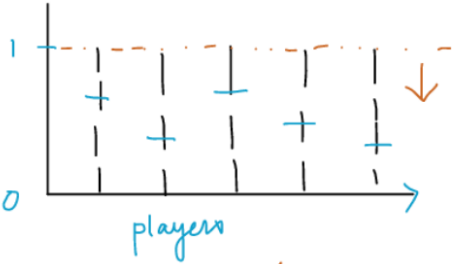
Uniform Rule (Sprumont 1991)

PE, ANON, and SP?

Question

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

- Suppose, $\sum_{i \in N} p_i < 1$
- Begin with everyone's allocation being 1 (infeasible), i.e., $f_i(P) = 1, \forall i \in N$
- Keep reducing until $\sum_{i \in N} f_i(P) = 1$
- On this path, if some agent's peak is reached, set the allocation for that agent to be its peak, i.e., $f_i(P) = p_i$



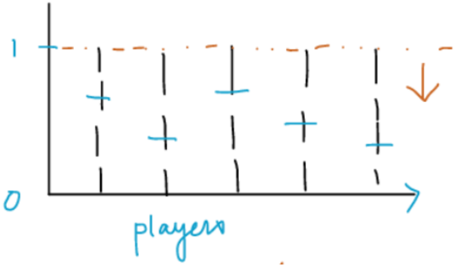
Uniform Rule (Sprumont 1991)

PE, ANON, and SP?

Question

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

- Suppose, $\sum_{i \in N} p_i < 1$
- Begin with everyone's allocation being 1 (infeasible), i.e., $f_i(P) = 1, \forall i \in N$
- Keep reducing until $\sum_{i \in N} f_i(P) = 1$
- On this path, if some agent's peak is reached, set the allocation for that agent to be its peak, i.e., $f_i(P) = p_i$
- Symmetric for $\sum_{i \in N} p_i > 1$

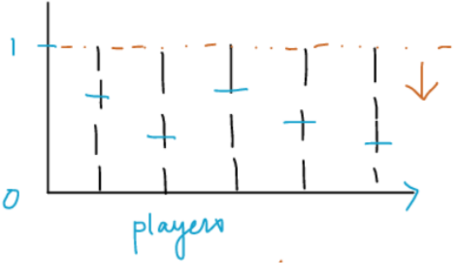


Uniform Rule (Sprumont 1991)

The Uniform Rule (Sprumont 1991)

Definition

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

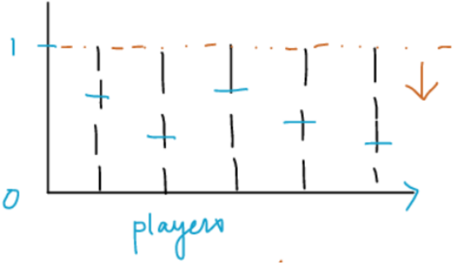


Uniform Rule (Sprumont 1991)

The Uniform Rule (Sprumont 1991)

Definition

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

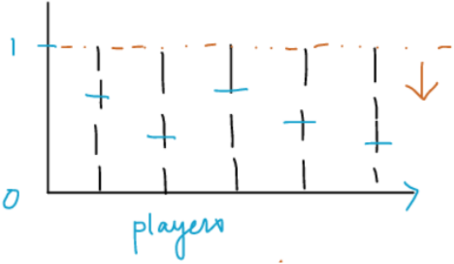


Uniform Rule (Sprumont 1991)

The Uniform Rule (Sprumont 1991)

Definition

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



Uniform Rule (Sprumont 1991)

The Uniform Rule

Theorem (Sprumont 1991)

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

The Uniform Rule

Theorem (Sprumont 1991)

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

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

The Uniform Rule

Theorem (Sprumont 1991)

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

- **ANON** is obvious: only the peaks matter and not their owners
- **PE**: the allocation is s.t.

The Uniform Rule

Theorem (Sprumont 1991)

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

- **ANON** is obvious: only the peaks matter and not their owners
- **PE**: the allocation is s.t.
  - $f_i^u(P) = p_i, \forall i \in N$, if $\sum_{i \in N} p_i = 1$

The Uniform Rule

Theorem (Sprumont 1991)

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

- **ANON** is obvious: only the peaks matter and not their owners
- **PE**: the allocation is s.t.
  - $f_i^u(P) = p_i, \forall i \in N$, if $\sum_{i \in N} p_i = 1$
  - $f_i^u(P) \geqslant p_i, \forall i \in N$, if $\sum_{i \in N} p_i < 1$

The Uniform Rule

Theorem (Sprumont 1991)

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

- **ANON** is obvious: only the peaks matter and not their owners
- **PE**: the allocation is s.t.
  - $f_i^u(P) = p_i, \forall i \in N$, if $\sum_{i \in N} p_i = 1$
  - $f_i^u(P) \geqslant p_i, \forall i \in N$, if $\sum_{i \in N} p_i < 1$
  - $f_i^u(P) \leqslant p_i, \forall i \in N$, if $\sum_{i \in N} p_i > 1$

The Uniform Rule

Theorem (Sprumont 1991)

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

- **ANON** is obvious: only the peaks matter and not their owners
- **PE**: the allocation is s.t.
  - $f_i^u(P) = p_i, \forall i \in N$, if $\sum_{i \in N} p_i = 1$
  - $f_i^u(P) \geqslant p_i, \forall i \in N$, if $\sum_{i \in N} p_i < 1$
  - $f_i^u(P) \leqslant p_i, \forall i \in N$, if $\sum_{i \in N} p_i > 1$
- This is PE from our previous observation on PE: *allocations should stay on the same side of the peaks for every agent*

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$
- The only way i can change the allocation is by reporting $p_i' > \mu(P) > p_i$

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$
- The only way i can change the allocation is by reporting $p_i' > \mu(P) > p_i$
- Leads to an worse outcome for i than $\mu(P)$

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$
- The only way i can change the allocation is by reporting $p_i' > \mu(P) > p_i$
- Leads to an worse outcome for i than $\mu(P)$
- A similar argument for case $\sum_{i \in N} p_i > 1$

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$
- The only way i can change the allocation is by reporting $p_i' > \mu(P) > p_i$
- Leads to an worse outcome for i than $\mu(P)$
- A similar argument for case $\sum_{i \in N} p_i > 1$

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$
- The only way i can change the allocation is by reporting $p_i' > \mu(P) > p_i$
- Leads to an worse outcome for i than $\mu(P)$
- A similar argument for case $\sum_{i \in N} p_i > 1$

The converse is also true, i.e.,

Theorem

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

- See **Sprumont (1991) : Division problem with single-peaked preferences**

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$
- The only way i can change the allocation is by reporting $p_i' > \mu(P) > p_i$
- Leads to an worse outcome for i than $\mu(P)$
- A similar argument for case $\sum_{i \in N} p_i > 1$

The converse is also true, i.e.,

Theorem

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

- See **Sprumont (1991) : Division problem with single-peaked preferences**
- **Envy-free (EF):** Agents do not envy each other's shares – also holds for uniform rule

The Uniform Rule: Strategyproofness

- **Case** $\sum_{i \in N} p_i = 1$: each agent gets her peak, no reason to deviate
- **Case** $\sum_{i \in N} p_i < 1$: then $f_i^u(P) \geqslant p_i, \forall i \in N$
- Manipulation, only for $i \in N$ s.t. $f_i^u(P) > p_i \implies \mu(P) > p_i$
- The only way i can change the allocation is by reporting $p_i' > \mu(P) > p_i$
- Leads to an worse outcome for i than $\mu(P)$
- A similar argument for case $\sum_{i \in N} p_i > 1$

The converse is also true, i.e.,

Theorem

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

- See **Sprumont (1991) : Division problem with single-peaked preferences**
- **Envy-free (EF):** Agents do not envy each other's shares – also holds for uniform rule
- **SP, PE, ANON, EF, polynomial-time computable**

Contents

- Task Allocation Domain
- The Uniform Rule
- Mechanism Design with Transfers
- Quasi Linear Preferences
- Pareto Optimality and Groves Payments

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$

Mechanism Design with Transfers

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

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \rightarrow X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \rightarrow X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:
    - **allocation** a

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:
  - **allocation** a
  - **payment** $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:
    - **allocation** a
    - **payment** $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \rightarrow X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:
  - **allocation** a
  - **payment** $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. A public decision to build a bridge, park, or museum. $a = \{\text{park, bridge}, \cdots\}$

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:
  - **allocation** a
  - **payment** $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. A public decision to build a bridge, park, or museum. $a = \{\text{park, bridge}, \cdots\}$
  2. Allocation of a divisible good, e.g., a shared spectrum, $a = (a_1, a_2, \cdots, a_n)$, $a_i \in [0, 1]$, $\sum_{i \in N} a_i = 1$, here a_i : fraction of the resource i gets

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:
  - **allocation** a
  - **payment** $\pi = (\pi_1, \cdots, \pi_n), \ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. A public decision to build a bridge, park, or museum. $a = \{\text{park}, \text{bridge}, \cdots\}$
  2. Allocation of a divisible good, e.g., a shared spectrum, $a = (a_1, a_2, \cdots, a_n)$, $a_i \in [0, 1]$, $\sum_{i \in N} a_i = 1$, here a_i : fraction of the resource i gets
  3. Single indivisible object allocation, e.g., a painting to be auctioned, $a = (a_1, a_2, \cdots, a_n)$, $a_i \in \{0, 1\}$, $\sum_{i \in N} a_i \leqslant 1$

Mechanism Design with Transfers

- Social Choice Function $F : \Theta \to X$
- X: space of all **outcomes**
- In this domain, an outcome $x \in X$ has two components:
  - **allocation** a
  - **payment** $\pi = (\pi_1, \cdots, \pi_n),\ \pi_i \in \mathbb{R}$
- Examples of allocations:
  1. A public decision to build a bridge, park, or museum. $a = \{\text{park, bridge}, \cdots\}$
  2. Allocation of a divisible good, e.g., a shared spectrum, $a = (a_1, a_2, \cdots, a_n)$, $a_i \in [0, 1]$, $\sum_{i \in N} a_i = 1$, here a_i : fraction of the resource i gets
  3. Single indivisible object allocation, e.g., a painting to be auctioned, $a = (a_1, a_2, \cdots, a_n)$, $a_i \in \{0, 1\}$, $\sum_{i \in N} a_i \leqslant 1$
  4. Partitioning indivisible objects, S = set of objects, $A = \{(A_1, \cdots, A_n) : A_i \subseteq S,\ \forall i \in N, A_i \cap A_j = \emptyset,\ \forall i \neq j\}$

Mechanism Design with Transfers

- Type of an agent i is $\theta_i \in \Theta_i$ this is a private information of i

Mechanism Design with Transfers

- Type of an agent i is $\theta_i \in \Theta_i$ this is a private information of i
- Agents *benefit* from an allocation is defined via the **valuation function**

Mechanism Design with Transfers

- Type of an agent i is $\theta_i \in \Theta_i$ this is a private information of i
- Agents *benefit* from an allocation is defined via the **valuation function**
- Valuation depends on the **allocation** and the **type** of the player

$$v_i : A \times \Theta_i \to \mathbb{R} \qquad \textbf{(independent private values)}$$

Mechanism Design with Transfers

- Type of an agent i is $\theta_i \in \Theta_i$ this is a private information of i
- Agents *benefit* from an allocation is defined via the **valuation function**
- Valuation depends on the **allocation** and the **type** of the player

$$v_i : A \times \Theta_i \to \mathbb{R} \qquad \textbf{(independent private values)}$$

- Examples:

Mechanism Design with Transfers

- Type of an agent i is $\theta_i \in \Theta_i$ this is a private information of i
- Agents *benefit* from an allocation is defined via the **valuation function**
- Valuation depends on the **allocation** and the **type** of the player

$$v_i : A \times \Theta_i \to \mathbb{R} \qquad \textbf{(independent private values)}$$

- Examples:
  - if i has a type ‘environmentalist’ θ_i^{env}, and $a \in \{\text{Bridge}, \text{Park}\}$, then $v_i(B, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$

Mechanism Design with Transfers

- Type of an agent i is $\theta_i \in \Theta_i$ this is a private information of i
- Agents *benefit* from an allocation is defined via the **valuation function**
- Valuation depends on the **allocation** and the **type** of the player

$$v_i : A \times \Theta_i \to \mathbb{R} \qquad \textbf{(independent private values)}$$

- Examples:
  - if i has a type 'environmentalist' θ_i^{env}, and $a \in \{\text{Bridge}, \text{Park}\}$, then $v_i(B, \theta_i^{\text{env}}) < v_i(P, \theta_i^{\text{env}})$
  - if type changes to 'business' θ_i^{bus}, $v_i(B, \theta_i^{\text{bus}}) > v_i(P, \theta_i^{\text{bus}})$

Payments = Monetary Transfers

- Unlike other domains, here we have an ‘instrument’ called **money** (also called **payment** or **transfers**)

Payments = Monetary Transfers

- Unlike other domains, here we have an ‘instrument’ called **money** (also called **payment** or **transfers**)
- **Payments** $\pi_i \in \mathbb{R}, \forall i \in N$

Payments = Monetary Transfers

- Unlike other domains, here we have an ‘instrument’ called **money** (also called **payment** or **transfers**)
- **Payments** $\pi_i \in \mathbb{R}, \forall i \in N$
- Payment vector $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$

Payments = Monetary Transfers

- Unlike other domains, here we have an 'instrument' called **money** (also called **payment** or **transfers**)
- **Payments** $\pi_i \in \mathbb{R}$, $\forall i \in N$
- Payment vector $\pi = (\pi_1, \pi_2, \dots, \pi_n)$
- Utility of player i, when its type is θ_i, and the outcome is $x = (a, \pi)$ is given by

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

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Question

Why is this a domain restriction?

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Question

Why is this a domain restriction?

Answer

- Consider two alternatives (a, π) and (a, π'), allocation is the same but payments are different

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Question

Why is this a domain restriction?

Answer

- Consider two alternatives (a, π) and (a, π'), allocation is the same but payments are different
- Suppose $\pi_i' < \pi_i$ for some $i \in N$

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Question

Why is this a domain restriction?

Answer

- Consider two alternatives (a, π) and (a, π'), allocation is the same but payments are different
- Suppose $\pi'_i < \pi_i$ for some $i \in N$
- There **cannot** be any preference profile in the quasi-linear domain where (a, π) is more preferred than (a, π') for agent i

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Question

Why is this a domain restriction?

Answer

- Consider two alternatives (a, π) and (a, π'), allocation is the same but payments are different
- Suppose $\pi'_i < \pi_i$ for some $i \in N$
- There **cannot** be any preference profile in the quasi-linear domain where (a, π) is more preferred than (a, π') for agent i
- Because $v_i(a,\theta_i)$ - $\pi'_i > v_i(a,\theta_i) - \pi_i, \forall \theta_i \in \Theta_i$

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Question

Why is this a domain restriction?

Answer

- Consider two alternatives (a, π) and (a, π'), allocation is the same but payments are different
- Suppose $\pi'_i < \pi_i$ for some $i \in N$
- There **cannot** be any preference profile in the quasi-linear domain where (a, π) is more preferred than (a, π') for agent i
- Because $v_i(a, \theta_i)$ - $\pi'_i > v_i(a, \theta_i) - \pi_i, \forall \theta_i \in \Theta_i$
- In the complete domain, both preference orders would have been feasible

Quasi Linear Domain

- Types θ_i that depend on the outcome $x = (a, \pi)$ this way belongs to the **quasi-linear domain**

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

Question

Why is this a domain restriction?

Answer

- Consider two alternatives (a, π) and (a, π'), allocation is the same but payments are different
- Suppose $\pi_i' < \pi_i$ for some $i \in N$
- There **cannot** be any preference profile in the quasi-linear domain where (a, π) is more preferred than (a, π') for agent i
- Because $v_i(a, \theta_i)$ - $\pi_i' > v_i(a,\theta_i) - \pi_i, \forall \theta_i \in \Theta_i$
- In the complete domain, both preference orders would have been feasible
- This restriction opens up possibilities of several non-dictatorial mechanisms

Contents

Quasi Linear preferences

- The SCF $F \equiv (f, (p_1, p_2, \dots, p_n)) \equiv (f, p)$ is decomposed into two components

Quasi Linear preferences

- The SCF $F \equiv (f, (p_1, p_2, \dots, p_n)) \equiv (f, p)$ is decomposed into two components
- **Allocation rule component**

$$f : \Theta_1 \times \Theta_2 \times \cdots \Theta_n \to A$$

When the types are $\theta_i, i \in N, f(\theta_1, \cdots, \theta_n) = a \in A$

Quasi Linear preferences

- The SCF $F \equiv (f, (p_1, p_2, \ldots, p_n)) \equiv (f, p)$ is decomposed into two components
- **Allocation rule component**

$$f : \Theta_1 \times \Theta_2 \times \cdots \Theta_n \rightarrow A$$

When the types are $\theta_i, i \in N, f(\theta_1, \cdots, \theta_n) = a \in A$

- **Payment function**

$$p_i : \Theta_1 \times \Theta_2 \times \cdots \Theta_n \rightarrow \mathbb{R}, \forall i \in N$$

When the types are $\theta_i, i \in N, p_i(\theta_1, \cdots, \theta_n) = \pi_i \in \mathbb{R}$

Example Allocation Rules

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

Example Allocation Rules

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

Example Allocation Rules

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

$$f^{AE}(\theta) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, \theta_i)$$

Note: This is different from Pareto efficiency (PE is a property defined for the outcome which also considers the payment)

Example Allocation Rules

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

$$f^{AE}(\theta) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, \theta_i)$$

   Note: This is different from Pareto efficiency (PE is a property defined for the outcome which also considers the payment)

4. **Affine maximizer rule**:

$$f^{AM}(\theta) \in \arg\max_{a \in A} (\sum_{i \in N} \lambda_i v_i(a, \theta_i) + \kappa(a)), \text{ where } \lambda_i \geqslant 0, \text{not all zero}$$

Example Allocation Rules

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

$$f^{AE}(\theta) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, \theta_i)$$

   Note: This is different from Pareto efficiency (PE is a property defined for the outcome which also considers the payment)

4. **Affine maximizer rule**:

$$f^{AM}(\theta) \in \arg\max_{a \in A} (\sum_{i \in N} \lambda_i v_i(a, \theta_i) + \kappa(a)), \text{ where } \lambda_i \geqslant 0, \text{ not all zero}$$

   - $\lambda_i = 1, \forall i \in N, \kappa \equiv 0$: allocatively efficient; $\lambda_d = 1, \lambda_j = 0, \forall j \in N \setminus \{d\}, \kappa \equiv 0$: dictatorial

Example Allocation Rules

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

$$f^{AE}(\theta) \in \arg\max_{a \in A} \sum_{i \in N} v_i(a, \theta_i)$$

   Note: This is different from Pareto efficiency (PE is a property defined for the outcome which also considers the payment)

4. **Affine maximizer rule**:

$$f^{AM}(\theta) \in \arg\max_{a \in A} \Big(\sum_{i \in N} \lambda_i v_i(a, \theta_i) + \kappa(a)\Big), \text{ where } \lambda_i \geqslant 0 \text{, not all zero}$$

   — $\lambda_i = 1, \forall i \in N, \kappa \equiv 0$: allocatively efficient; $\lambda_d = 1, \lambda_j = 0, \forall j \in N \setminus \{d\}, \kappa \equiv 0$: dictatorial

5. **Max-min/egalitarian**

$$f^{MM}(\theta) \in \arg\max_{a \in A} \min_{i \in N} v_i(a, \theta_i)$$

Example Payment Rules

1. **No deficit:** $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$

Example Payment Rules

1. **No deficit**: $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
2. **No subsidy**: $p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta, \forall i \in N$

Example Payment Rules

1. **No deficit**: $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
2. **No subsidy**: $p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta, \forall i \in N$
3. **Budget balanced**: $\sum_{i \in N} p_i(\theta) = 0, \ \forall \theta \in \Theta$

Example Payment Rules

1. **No deficit**: $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
2. **No subsidy**: $p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta, \forall i \in N$
3. **Budget balanced**: $\sum_{i \in N} p_i(\theta) = 0, \ \forall \theta \in \Theta$

Example Payment Rules

1. **No deficit:** $\sum_{i \in N} p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta$
2. **No subsidy:** $p_i(\theta) \geqslant 0, \ \forall \theta \in \Theta, \forall i \in N$
3. **Budget balanced:** $\sum_{i \in N} p_i(\theta) = 0, \ \forall \theta \in \Theta$

Definition (DSIC)

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

$$v_i(f(\theta_i, \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i, \tilde{\theta}_{-i}) \geqslant v_i(f(\theta_i', \tilde{\theta}_{-i}), \theta_i) - p_i(\theta_i', \tilde{\theta}_{-i}), \forall \tilde{\theta}_{-i} \in \Theta_{-i}, \theta_i', \theta_i \in \Theta_i, \forall i \in N$$

DSIC

- DSIC means truthtelling is a weakly DSE

DSIC

- DSIC means truthtelling is a weakly DSE
- We say that the payment rule p implements an allocation rule f in dominant strategies (OR) f is implementable in dominant strategies (by a payment rule)

DSIC

- DSIC means truthtelling is a weakly DSE
- We say that the payment rule p implements an allocation rule f in dominant strategies (OR) f is implementable in dominant strategies (by a payment rule)
- In QL domain, we are often more interested in the allocation rule than the whole SCF (which also includes payment)

DSIC

- DSIC means truthtelling is a weakly DSE
- We say that the payment rule p implements an allocation rule f in dominant strategies (OR) f is implementable in dominant strategies (by a payment rule)
- In QL domain, we are often more interested in the allocation rule than the whole SCF (which also includes payment)

DSIC

- DSIC means truthtelling is a weakly DSE
- We say that the payment rule p implements an allocation rule f in dominant strategies (OR) f is implementable in dominant strategies (by a payment rule)
- In QL domain, we are often more interested in the allocation rule than the whole SCF (which also includes payment)

Question

What needs to be satisfied for a DSIC mechanism (f, p)?

Implications of DSIC

Question

What needs to be satisfied for a DSIC mechanism (f, p)?

Implications of DSIC

Question

What needs to be satisfied for a DSIC mechanism (f, p)?

Example

$N = \{1, 2\}, \Theta_1 = \Theta_2 = \{\theta^H, \theta^L\}, f : \Theta_1 \times \Theta_2 \to A$. The following conditions must hold

Implications of DSIC

Question

What needs to be satisfied for a DSIC mechanism (f, p)?

Example

$N = \{1, 2\}, \Theta_1 = \Theta_2 = \{\theta^H, \theta^L\}, f : \Theta_1 \times \Theta_2 \to A$. The following conditions must hold

Player 1:

$$v_1(f(\theta^H, \theta_2), \theta^H) - p_1(\theta^H, \theta_2) \geqslant v_1(f(\theta^L, \theta_2), \theta^H) - p_1(\theta^L, \theta_2), \forall \theta_2 \in \Theta_2 \quad (1)$$

$$v_1(f(\theta^L, \theta_2), \theta^L) - p_1(\theta^L, \theta_2) \geqslant v_1(f(\theta^H, \theta_2), \theta^L) - p_1(\theta^H, \theta_2), \forall \theta_2 \in \Theta_2 \quad (2)$$

Implications of DSIC

Question

What needs to be satisfied for a DSIC mechanism (f, p)?

Example

$N = \{1,2\}, \Theta_1 = \Theta_2 = \{\theta^H, \theta^L\}, f : \Theta_1 \times \Theta_2 \to A$. The following conditions must hold

Player 1:

$$v_1(f(\theta^H, \theta_2), \theta^H) - p_1(\theta^H, \theta_2) \geqslant v_1(f(\theta^L, \theta_2), \theta^H) - p_1(\theta^L, \theta_2), \forall \theta_2 \in \Theta_2 \tag{1}$$

$$v_1(f(\theta^L, \theta_2), \theta^L) - p_1(\theta^L, \theta_2) \geqslant v_1(f(\theta^H, \theta_2), \theta^L) - p_1(\theta^H, \theta_2), \forall \theta_2 \in \Theta_2 \tag{2}$$

Player 2:

$$v_2(f(\theta^H, \theta_1), \theta^H) - p_2(\theta^H, \theta_1) \geqslant v_2(f(\theta^L, \theta_1), \theta^H) - p_2(\theta^L, \theta_1), \forall \theta_1 \in \Theta_1 \tag{3}$$

$$v_2(f(\theta^L, \theta_1), \theta^L) - p_2(\theta^L, \theta_1) \geqslant v_2(f(\theta^H, \theta_1), \theta^L) - p_2(\theta^H, \theta_1), \forall \theta_1 \in \Theta_1 \tag{4}$$

Properties of the Payment

- Say (f, p) is incentive compatible, i.e., p implements f

Properties of the Payment

- Say (f, p) is incentive compatible, i.e., p implements f
- Consider another payment

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

Properties of the Payment

- Say (f, p) is incentive compatible, i.e., p implements f
- Consider another payment

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

- Question: Is (f, q) DSIC?

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

Properties of the Payment

- Say (f, p) is incentive compatible, i.e., p implements f
- Consider another payment

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

- Question: Is (f, q) DSIC?

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

- If we can find a payment that implements an allocation rule, there exists uncountably many payments that can implement it

Properties of the Payment

- Say (f, p) is incentive compatible, i.e., p implements f
- Consider another payment

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

- Question: Is (f, q) DSIC?

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

- If we can find a payment that implements an allocation rule, there exists uncountably many payments that can implement it
- **The converse question:** when do the payments that implement f differ only by a factor $h_i(\theta_{-i})$?

Properties of the Payment

- Suppose the allocation is same in two type profiles θ and $\tilde{\theta} = (\tilde{\theta}_i, \theta_{-i})$
- i.e., $f(\theta) = f(\tilde{\theta}) = a$, then
- if p implements f, then $p_i(\theta) = p_i(\tilde{\theta})$ **[exercise]**

Contents

- Task Allocation Domain
- The Uniform Rule
- Mechanism Design with Transfers
- Quasi Linear Preferences
- Pareto Optimality and Groves Payments

Pareto Optimality in Quasi-linear domain

Definition (Pareto Optimal)

A mechanism $(f, (p_1, \dots, p_n))$ is **Pareto optimal** if for every type profile $\theta \in \Theta$, there does not exist an allocation $b \neq f(\theta)$ and payments $(\pi_1, \dots, \pi_n)$ s.t.,

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

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

Pareto Optimality in Quasi-linear domain

Definition (Pareto Optimal)

A mechanism $(f, (p_1, \dots, p_n))$ is **Pareto optimal** if for every type profile $\theta \in \Theta$, there does not exist an allocation $b \neq f(\theta)$ and payments $(\pi_1, \dots, \pi_n)$ with $\sum_{i \in N} \pi_i \geqslant \sum_{i \in N} p_i(\theta)$ s.t.,

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

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

- Pareto optimality is meaningless if there is no restriction on the payment
- One can always put excessive subsidy to every agent to make everyone better off
- So, the condition requires to spend at least the same budget

Pareto Optimality in Quasi-linear Domain

Theorem

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

Pareto Optimality in Quasi-linear Domain

Theorem

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

- $(\Longleftarrow)$ we prove $\neg$PO $\implies$ $\neg$AE

Pareto Optimality in Quasi-linear Domain

Theorem

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

- ($\Longleftarrow$) we prove $\neg$PO $\implies$ $\neg$AE
- $\neg$PO, $\exists b, \pi, \theta$ s.t. $\sum_{i \in N} \pi_i \geqslant \sum_{i \in N} p_i(\theta)$

Pareto Optimality in Quasi-linear Domain

Theorem

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

- ($\Longleftarrow$) we prove $\neg$PO $\implies$ $\neg$AE
- $\neg$PO, $\exists b, \pi, \theta$ s.t. $\sum_{i \in N} \pi_i \geqslant \sum_{i \in N} p_i(\theta)$
- $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta), \forall i \in N$, strict for some $j \in N$

Pareto Optimality in Quasi-linear Domain

Theorem

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

- $(\impliedby)$ we prove $\neg$PO $\implies$ $\neg$AE
- $\neg$PO, $\exists b, \pi, \theta$ s.t. $\sum_{i \in N} \pi_i \geqslant \sum_{i \in N} p_i(\theta)$
- $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta), \forall i \in N$, strict for some $j \in N$
- summing over the all these inequalities

$$\sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} \pi_i > \sum_{i \in N} v_i(f(\theta), \theta_i) - \sum_{i \in N} p_i(\theta)$$
$$\sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} v_i(f(\theta), \theta_i) > \sum_{i \in N} \pi_i - \sum_{i \in N} p_i(\theta) \geqslant 0$$

Pareto Optimality in Quasi-linear Domain

Theorem

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

- $(\Longleftarrow)$ we prove $\neg$PO $\implies$ $\neg$AE
- $\neg$PO, $\exists b, \pi, \theta$ s.t. $\sum_{i \in N} \pi_i \geqslant \sum_{i \in N} p_i(\theta)$
- $v_i(b, \theta_i) - \pi_i \geqslant v_i(f(\theta), \theta_i) - p_i(\theta), \forall i \in N$, strict for some $j \in N$
- summing over the all these inequalities

$$\sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} \pi_i > \sum_{i \in N} v_i(f(\theta), \theta_i) - \sum_{i \in N} p_i(\theta)$$

$$\sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} v_i(f(\theta), \theta_i) > \sum_{i \in N} \pi_i - \sum_{i \in N} p_i(\theta) \geqslant 0$$

- f is $\neg$AE

Proof (contd.)

- $(\implies)$ $\neg$AE $\implies$ $\neg$PO

Proof (contd.)

- $(\implies)\ \neg\text{AE} \implies \neg\text{PO}$
- $\neg\text{AE} \implies \exists\theta, b \neq f(\theta)$ s.t. $\sum_{i\in N} v_i(b,\theta_i) > \sum_{i\in N} v_i(f(\theta),\theta_i)$

Proof (contd.)

- $(\implies)$ $\neg$AE $\implies$ $\neg$PO
- $\neg$AE $\implies$ $\exists \theta, b \neq f(\theta)$ s.t. $\sum_{i \in N} v_i(b, \theta_i) > \sum_{i \in N} v_i(f(\theta), \theta_i)$
- Let $\delta = \sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} v_i(f(\theta), \theta_i) > 0$

Proof (contd.)

- ($\implies$) $\neg$AE $\implies \neg$PO
- $\neg$AE $\implies \exists \theta, b \neq f(\theta)$ s.t. $\sum_{i \in N} v_i(b, \theta_i) > \sum_{i \in N} v_i(f(\theta), \theta_i)$
- Let $\delta = \sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} v_i(f(\theta), \theta_i) > 0$
- Consider payment $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n, \forall i \in N$

Proof (contd.)

- ($\implies$) $\neg$AE $\implies$ $\neg$PO
- $\neg$AE $\implies$ $\exists \theta, b \neq f(\theta)$ s.t. $\sum_{i \in N} v_i(b, \theta_i) > \sum_{i \in N} v_i(f(\theta), \theta_i)$
- Let $\delta = \sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} v_i(f(\theta), \theta_i) > 0$
- Consider payment $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta / n, \forall i \in N$
- Hence, $(v_i(b, \theta_i) - \pi_i) - (v_i(f(\theta), \theta_i) - p_i(\theta)) = \delta / n > 0, \forall i \in N$

Proof (contd.)

- ($\implies$) $\neg$AE $\implies$ $\neg$PO
- $\neg$AE $\implies$ $\exists \theta, b \neq f(\theta)$ s.t. $\sum_{i \in N} v_i(b, \theta_i) > \sum_{i \in N} v_i(f(\theta), \theta_i)$
- Let $\delta = \sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} v_i(f(\theta), \theta_i) > 0$
- Consider payment $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n, \forall i \in N$
- Hence, $(v_i(b, \theta_i) - \pi_i) - (v_i(f(\theta), \theta_i) - p_i(\theta)) = \delta/n > 0, \forall i \in N$
- also $\sum_{i \in N} \pi_i = \sum_{i \in N} p_i(\theta)$

Proof (contd.)

- ($\implies$) $\neg$AE $\implies \neg$PO
- $\neg$AE $\implies \exists \theta, b \neq f(\theta)$ s.t. $\sum_{i \in N} v_i(b, \theta_i) > \sum_{i \in N} v_i(f(\theta), \theta_i)$
- Let $\delta = \sum_{i \in N} v_i(b, \theta_i) - \sum_{i \in N} v_i(f(\theta), \theta_i) > 0$
- Consider payment $\pi_i = v_i(b, \theta_i) - v_i(f(\theta), \theta_i) + p_i(\theta) - \delta/n, \forall i \in N$
- Hence, $(v_i(b, \theta_i) - \pi_i) - (v_i(f(\theta), \theta_i) - p_i(\theta)) = \delta/n > 0, \forall i \in N$
- also $\sum_{i \in N} \pi_i = \sum_{i \in N} p_i(\theta)$
- Hence f is not PO

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Example

- Single indivisible item allocation $N = \{1, 2, 3, 4\}$

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Example

- Single indivisible item allocation $N = \{1, 2, 3, 4\}$
- $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$, when they get the object, zero otherwise

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Example

- Single indivisible item allocation $N = \{1, 2, 3, 4\}$
- $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$, when they get the object, zero otherwise
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Example

- Single indivisible item allocation $N = \{1, 2, 3, 4\}$
- $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$, when they get the object, zero otherwise
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$
- If everyone reports their true type, the values of h_i are $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Example

- Single indivisible item allocation $N = \{1, 2, 3, 4\}$
- $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$, when they get the object, zero otherwise
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$
- If everyone reports their true type, the values of h_i are $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$
- The efficient allocation gives the item to agent 1

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Example

- Single indivisible item allocation $N = \{1, 2, 3, 4\}$
- $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$, when they get the object, zero otherwise
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$
- If everyone reports their true type, the values of h_i are $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$
- The efficient allocation gives the item to agent 1
- $p_1 = 4 - 0 = 4, p_2 = 4 - 10 = -6, p_3 = 4 - 10 = -6, p_4 = 6 - 10 = -4$, i.e., only player 1 pays, other get paid

Allocatively Efficient Rule is Implementable

- Consider the following payment: $p_i^G(\theta_i, \theta_{-i}) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(f^{AE}(\theta_i, \theta_{-i}), \theta_j)$, where $h_i : \Theta_{-i} \to \mathbb{R}$ is an arbitrary function: **Groves payment**

Example

- Single indivisible item allocation $N = \{1, 2, 3, 4\}$
- $\theta_1 = 10$, $\theta_2 = 8$, $\theta_3 = 6$, $\theta_4 = 4$, when they get the object, zero otherwise
- Let $h_i(\theta_{-i}) = \min \theta_{-i}$
- If everyone reports their true type, the values of h_i are $h_1 = 4$, $h_2 = 4$, $h_3 = 4$, $h_4 = 6$
- The efficient allocation gives the item to agent 1
- $p_1 = 4 - 0 = 4, p_2 = 4 - 10 = -6, p_3 = 4 - 10 = -6, p_4 = 6 - 10 = -4$, i.e., only player 1 pays, other get paid
- **Surprisingly, this is a truthful mechanism**

Groves mechanisms are Truthful

Theorem

Groves mechanisms are DSIC

- Consider player i

Groves mechanisms are Truthful

Theorem

Groves mechanisms are DSIC

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

Groves mechanisms are Truthful

Theorem

Groves mechanisms are DSIC

- Consider player i
- $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$, and $f^{AE}(\theta_i', \tilde{\theta}_{-i}) = b$
- By definition, $v_i(a, \theta_i) + \sum_{j \neq i} v_j(a, \tilde{\theta}_j) \geqslant v_i(b, \theta_i) + \sum_{j \neq i} v_j(b, \tilde{\theta}_j)$

Groves mechanisms are Truthful

Theorem

Groves mechanisms are DSIC

- Consider player i
- $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$, and $f^{AE}(\theta'_i, \tilde{\theta}_{-i}) = b$
- By definition, $v_i(a, \theta_i) + \sum_{j \neq i} v_j(a, \tilde{\theta}_j) \geqslant v_i(b, \theta_i) + \sum_{j \neq i} v_j(b, \tilde{\theta}_j)$
- utility of player i when he reports θ_i is

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

Groves mechanisms are Truthful

Theorem

Groves mechanisms are DSIC

- Consider player i
- $f^{AE}(\theta_i, \tilde{\theta}_{-i}) = a$, and $f^{AE}(\theta'_i, \tilde{\theta}_{-i}) = b$
- By definition, $v_i(a, \theta_i) + \sum_{j \neq i} v_j(a, \tilde{\theta}_j) \geqslant v_i(b, \theta_i) + \sum_{j \neq i} v_j(b, \tilde{\theta}_j)$
- utility of player i when he reports θ_i is

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

- Since player i was arbitrary, this holds for all $i \in N$. Hence the claim.

भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay