Topics in Computational Social Choice Theory

Lecture 01: Introduction on Discrete Fair Division

Hannaneh Akrami
Organization

Seminar: 2+0, 7 CPS

Organized by Kurt Mehlhorn, Nidhi Rathi, and Hannaneh Akrami

When? Every Tuesday 14:15 - 15:45

Requirements: Basic algorithms lecture
(Introduction to Algorithms and Data Structures)

Your task:
• Present a paper from the list in 60-85 minutes.
• Write a summary of the paper by August 2nd.
• The presentation needs to be discussed with us at least one week before your scheduled talk.
• Send us your preferred order of the papers by April 30th.
Computational Social Choice Theory

Social Choice Theory: Making a collective decision from individual preferences.
Computational Social Choice Theory

Social Choice Theory: Making a collective decision from individual preferences.

Voting
Social Choice Theory: Making a collective decision from individual preferences.

Voting

Resource Allocation
Computational Social Choice Theory

Social Choice Theory: Making a collective decision from individual preferences.

Voting

Resource Allocation

Stable Matchings
Computational Social Choice Theory

Social Choice Theory: Making a collective decision from individual preferences.

Economists and Politicians: Does there exist a social choice mechanism with the desired economic properties?
Computational Social Choice Theory

Social Choice Theory: Making a collective decision from individual preferences.

Economists and Politicians: Does there exist a social choice mechanism with the desired economic properties?

Computer Scientists: How to efficiently compute such a mechanism?
Computational Social Choice Theory

**Social Choice Theory:** Making a collective decision from individual preferences.

**Economists and Politicians:** Does there exists a social choice mechanism with the desired economic properties?

**Computer Scientists:** How to efficiently compute such a mechanism?
Fair Division

Divide **items** among **agents** in a **fair** manner.
Fair Division

Divide **items** among **agents** in a **fair** manner.

Applications:

- Partnership dissolution
- Divorce settlements
- Household chores
- Air traffic management
Items

Desirable

Undesirable
Items

Desirable

Divorce settlements

Undesirable

Household chores
Items

Desirable

Divisible goods

Indivisible goods

Undesirable
Items

Desirable

Divisible goods

Indivisible goods

Undesirable
Items

Desirable
- Divisible goods
- Indivisible goods

Undesirable
- Divisible chores
- Indivisible chores
Items

- Desirable
  - Divisible goods
  - Indivisible goods
- Undesirable
  - Divisible chores
  - Indivisible chores
Items

Desirable

Divisible goods

Indivisible goods

Undesirable

Divisible chores

Indivisible chores

Today
Items

Desirable
- Divisible goods
- Next week
- Today

Undesirable
- Undesirable
- Divisible chores
- Indivisible chores
- Next week
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

Input: $\mathcal{I} = (N, M, V)$

- $N$: set of $n$ agents
- $M$: set of $m$ indivisible goods
- Valuation functions $v_i : 2^M \to \mathbb{R}_{\geq 0}$
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

**Input:** \( \mathcal{I} = (N, M, V) \)

- \( N \): set of \( n \) agents
- \( M \): set of \( m \) indivisible goods
- Valuation functions \( v_i : 2^M \to \mathbb{R}_{\geq 0} \)
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

**Input:** $\mathcal{I} = (N, M, V)$

- $N$: set of $n$ agents
- $M$: set of $m$ indivisible goods
- Valuation functions $v_i : 2^M \rightarrow \mathbb{R}_{\geq 0}$
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

**Input:** \(I = (N, M, V)\)

- \(N\): set of \(n\) agents
- \(M\): set of \(m\) indivisible goods
- Valuation functions \(v_i : 2^M \rightarrow \mathbb{R}_{\geq 0}\)

<table>
<thead>
<tr>
<th></th>
<th>🍎</th>
<th>🍊</th>
<th>🍓</th>
<th>🍊</th>
<th>🍊</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍓</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>🍓</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>🍓</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
**Discrete Fair Division**

Divide *indivisible items* among *agents* in a *fair* manner.

**Input:** \( I = (N, M, V) \)

- \( N \): set of \( n \) agents
- \( M \): set of \( m \) indivisible goods
- Valuation functions \( v_i : 2^M \rightarrow \mathbb{R}_{\geq 0} \)

<table>
<thead>
<tr>
<th></th>
<th>( \text{Banana} )</th>
<th>( \text{Apple} )</th>
<th>( \text{Strawberry} )</th>
<th>( \text{Orange} )</th>
<th>( \text{Peach} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent 1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Agent 2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Agent 3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Goal:** Find a *fair* allocation of the goods to the agents.
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

Input: $\mathcal{I} = (N, M, V)$

- $N$: set of $n$ agents
- $M$: set of $m$ indivisible goods
- Valuation functions $v_i : 2^M \rightarrow \mathbb{R}_{\geq 0}$

Goal: Find a **fair** allocation of the goods to the agents.

A partition $X = (X_1, X_2, \ldots, X_n, P)$ of $M$
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

**Input:** \( I = (N, M, V) \)

- **\( N \):** set of \( n \) agents
- **\( M \):** set of \( m \) indivisible goods
- **Valuation functions** \( v_i : 2^M \rightarrow \mathbb{R}_{\geq 0} \)

**Goal:** Find a **fair** allocation of the goods to the agents.

A partition \( X = (X_1, X_2, \ldots, X_n, P) \) of \( M \)
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

Input: \( I = (N, M, V) \)

- \( N = \{a_1, a_2, a_3\} \)
- \( M = \{g_1, g_2, g_3, g_4, g_5\} \)
- \( X_1 = \{g_1\}, X_2 = \{g_2, g_5\}, X_3 = \{g_3\}, P = \{g_4\} \)
- \( v_1(X_1) = 4, v_1(X_2) = 3 \)
## Discrete Fair Division

Divide *indivisible items* among *agents* in a *fair* manner.

**Input:** $\mathcal{I} = (N, M, V)$

- $N = \{a_1, a_2, a_3\}$
- $M = \{g_1, g_2, g_3, g_4, g_5\}$
- $X_1 = \{g_1\}$, $X_2 = \{g_2, g_5\}$, $X_3 = \{g_3\}$, $P = \{g_4\}$
- $v_1(X_1) = 4$, $v_1(X_2) = 3$

<table>
<thead>
<tr>
<th></th>
<th>$g_1$</th>
<th>$g_2$</th>
<th>$g_3$</th>
<th>$g_4$</th>
<th>$g_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$a_2$</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$a_3$</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

*assuming $v_1$ is additive: for all $S \subseteq M$, $v_1(S) = \sum_{g \in S} v_i(\{g\})$*
Discrete Fair Division

Divide **indivisible items** among **agents** in a **fair** manner.

**Input:** \( \mathcal{I} = (N, M, V) \)

- \( N = \{a_1, a_2, a_3\} \)
- \( M = \{g_1, g_2, g_3, g_4, g_5\} \)
- \( X_1 = \{g_1\}, \ X_2 = \{g_2, g_5\}, \ X_3 = \{g_3\}, \ P = \{g_4\} \)
- \( v_1(X_1) = 4, \ v_1(X_2) = 3 \)

<table>
<thead>
<tr>
<th></th>
<th>( g_1 )</th>
<th>( g_2 )</th>
<th>( g_3 )</th>
<th>( g_4 )</th>
<th>( g_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( a_3 )</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

An allocation is **complete**, if \( P = \emptyset \) and **partial** otherwise.
Fairness
Which allocation is fair?
Fairness

- Envy Based
- Share Based
Fairness

- Envy Based
  - Envy Freeness
  - EF1
  - EFX
- Share Based
Fairness

- Envy Based
  - Envy Freeness
  - EF1
  - EFX
- Share Based
  - Proportionality
  - MMS
Envy Freeness

**Definition:** An allocation $X$ is **envy free**, if and only if for all agents $a_i, a_j$:

$v_i(X_i) \geq v_i(X_j)$.  [Foley 1967]
**Envy Freeness**

**Definition:** An allocation $X$ is **envy free**, if and only if for all agents $a_i, a_j$:
$$v_i(X_i) \geq v_i(X_j).$$ [Foley 1967]

Which allocation is envy free?

1.  
2.  
3.  

[Diagrams showing allocations]
Envy Freeness

**Definition:** An allocation $X$ is envy free, if and only if for all agents $a_i, a_j:$
$v_i(X_i) \geq v_i(X_j).$ [Foley 1967]

Do complete envy free allocations always exist?
Envy Freeness

Definition: An allocation $X$ is envy free, if and only if for all agents $a_i, a_j$:
$\forall i \in [1,n], \forall j \in [1,n] (i \neq j): v_i(X_i) \geq v_i(X_j)$. [Foley 1967]

Do complete envy free allocations always exist?

- For divisible goods, YES! (Next weeks)
Envy Freeness

Definition: An allocation $X$ is envy free, if and only if for all agents $a_i, a_j$:

$$v_i(X_i) \geq v_i(X_j).$$  [Foley 1967]

Do complete envy free allocations always exist?

- For divisible goods, YES! (Next weeks)
- For indivisible goods, NO!

Envy Freeness

**Definition:** An allocation $X$ is **envy free**, if and only if for all agents $a_i, a_j$:

$$v_i(X_i) \geq v_i(X_j).$$

[Foley 1967]

Do complete envy free allocations always exist?

- For divisible goods, YES! (Next weeks)
- For indivisible goods, NO!

Others should not get more than me!
**EF1**

**Definition:** An allocation $X$ is **envy free up to one item** or **EF1**, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$. 
**EF1**

**Definition:** An allocation $X$ is **envy free up to one item** or EF1, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$. 
EF1

Definition: An allocation $X$ is envy free up to one item or EF1, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

I do not envy him if the apple is removed!
**EF1**

**Definition:** An allocation $X$ is **envy free up to one item** or EF1, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

Do complete EF1 allocations always exist?
**EF1**

**Definition:** An allocation $X$ is **envy free up to one item** or **EF1**, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

Do complete EF1 allocations always exist?

- YES for monotone valuations!

I do not envy him if the apple is removed!
EF1

Definition: An allocation $X$ is envy free up to one item or EF1, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

Do complete EF1 allocations always exist?

- YES for monotone valuations!

for all $S \subseteq M$ and $g \in M$, $v(S \cup \{g\}) \geq v(S)$
**EF1**

**Definition:** An allocation $X$ is **envy free up to one item** or **EF1**, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

Do complete EF1 allocations always exist?

- **YES** for **monotone valuations**!

  for all $S \subseteq M$ and $g \in M$, $v(S \cup \{g\}) \geq v(S)$

- A complete EF1 allocation can be found in polynomial time.

  [Lipton, Markakis, Mossel, Saberi 2004]
**EF1**

**Definition:** An allocation $X$ is envy free up to one item or EF1, if and only if for all agents $a_i, a_j$, there exists a good $g \in X_j$ (if $X_j \neq \emptyset$): $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

Do complete EF1 allocations always exist?

- YES for monotone valuations!

> for all $S \subseteq M$ and $g \in M$, $v(S \cup \{g\}) \geq v(S)$

- A complete EF1 allocation can be found in polynomial time.

- Today: A polynomial time algorithm to find a complete EF1 allocation for additive valuations.

[Lipton, Markakis, Mossel, Saberi 2004]
Round-Robin Algorithm

• Fix an ordering of the agents, say $a_1, a_2, \ldots, a_n$.
• Agents take turns according to the ordering $(a_1, a_2, \ldots, a_n, a_1, a_2, \ldots, a_n, \ldots)$ to pick their favorite items from the set of the remaining items.
Round-Robin Algorithm

- Fix an ordering of the agents, say $a_1, a_2, \ldots, a_n$.
- Agents take turns according to the ordering $(a_1, a_2, \ldots, a_n, a_1, a_2, \ldots, a_n, \ldots)$ to pick their favorite items from the set of the remaining items.

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
<th>🍓</th>
<th>🍊</th>
<th>🍊</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧑‍♀️</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>🧑‍♀️</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>🧑‍♀️</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Round-Robin Algorithm

- Fix an ordering of the agents, say $a_1, a_2, \ldots, a_n$.
- Agents take turns according to the ordering $(a_1, a_2, \ldots, a_n, a_1, a_2, \ldots, a_n, \ldots)$ to pick their favorite items from the set of the remaining items.

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
<th>🍓</th>
<th>🍊</th>
<th>🍉</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Round-Robin Algorithm

- Fix an ordering of the agents, say \(a_1, a_2, \ldots, a_n\).
- Agents take turns according to the ordering \((a_1, a_2, \ldots, a_n, a_1, a_2, \ldots, a_n, \ldots)\) to pick their favorite items from the set of the remaining items.

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
<th>🍓</th>
<th>🍊</th>
<th>🍊</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧑‍ê�</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>🧑‍ู่</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>🧑‍ ريال</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Round-Robin Algorithm

- Fix an ordering of the agents, say $a_1, a_2, \ldots, a_n$.
- Agents take turns according to the ordering $(a_1, a_2, \ldots, a_n, a_1, a_2, \ldots, a_n, \ldots)$ to pick their favorite items from the set of the remaining items.

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
<th>🍓</th>
<th>🍊</th>
<th>🍏</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧑‍❤️‍👨</td>
<td>[4]</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>🧑‍:Any</td>
<td>1</td>
<td>0</td>
<td>[5]</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>🧑‍:Any</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>[1]</td>
</tr>
</tbody>
</table>
Round-Robin Algorithm

- Fix an ordering of the agents, say $a_1, a_2, \ldots, a_n$.
- Agents take turns according to the ordering $(a_1, a_2, \ldots, a_n, a_1, a_2, \ldots, a_n, \ldots)$ to pick their favorite items from the set of the remaining items.

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
<th>🍓</th>
<th>🍊</th>
<th>🍏</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Round-Robin Algorithm

- Fix an ordering of the agents, say $a_1, a_2, \ldots, a_n$.
- Agents take turns according to the ordering $(a_1, a_2, \ldots, a_n, a_1, a_2, \ldots, a_n, \ldots)$ to pick their favorite items from the set of the remaining items.
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

\[ a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_n \]
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

\[ a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_n \]

First round:
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

\[ a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_n \]

First round: \[ \bullet \]
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.
**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

\[ a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_n \]

First round: [Image of colored circles indicating allocation]
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

\[ a_1, a_2, a_3, \ldots, a_n \]

First round: \[ \bullet, \odot, \odot, \cdots, \bullet \]

Second round: \[ \bullet \]
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

\[a_1, a_2, a_3, \ldots, a_n\]

First round: 
- \(a_1\) (red)
- \(a_2\) (yellow)
- \(a_3\) (blue)
- \(\ldots\)
- \(a_n\) (black)

Second round: 
- \(a_1\) (red)
- \(a_2\) (yellow)
- \(a_3\) (blue)
- \(\ldots\)
- \(a_n\) (black)
Round-Robin Algorithm

Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

\[
\begin{array}{cccccc}
\ldots & a_1 & a_2 & a_3 & \ldots & a_n \\
\text{First round:} & \bullet & \circ & \bullet & \ldots & \bullet \\
\text{Second round:} & \bullet & \circ & \bullet & \ldots & \bullet \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\text{Last round:} & \end{array}
\]
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r, b)\). Analyze envy from \(r\) to \(b\).
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r, b)\). Analyze envy from \(r\) to \(b\).

\[
\begin{array}{c}
\text{First round:} \\
\text{Second round:} \\
\text{Last round:}
\end{array}
\]

\[
\begin{array}{c}
\cdots \, \bullet \, \cdots \, \bullet \, \cdots \\
\cdots \, \bullet \, \cdots \, \bullet \, \cdots \\
\cdots \, \cdots \, \cdots \\
\cdots \, \bullet
\end{array}
\]
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r, b)\). Analyze envy from \(r\) to \(b\).

\[
\begin{align*}
\text{First round:} & \quad \cdots \quad r \rightarrow b \quad \cdots \\
\text{Second round:} & \quad \cdots \quad r \rightarrow b \quad \cdots \\
\text{Last round:} & \quad \cdots \quad b
\end{align*}
\]
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r, b)\). Analyze envy from \(r\) to \(b\).

If \(r\) precedes \(b\), by additivity
\[
v_r(X_r) \geq v_r(X_b).
\]
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r, b)\). Analyze envy from \(r\) to \(b\).
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r, b)\). Analyze envy from \(r\) to \(b\).

\[
\begin{array}{c|c|c}
 & b & r \\
\hline
\text{First round:} & \cdots & \bullet & \cdots & \bullet & \cdots \\
\text{Second round:} & \cdots & \bullet & \cdots & \bullet & \cdots \\
\hline
\text{Last round:} & \cdots & \bullet
\end{array}
\]
Theorem: For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents $(r, b)$. Analyze envy from $r$ to $b$.

\begin{align*}
\text{First round:} & \quad \cdots \quad g \quad \cdots \quad \bullet \quad \cdots \\
\text{Second round:} & \quad \cdots \quad \bullet \quad \cdots \quad \bullet \quad \cdots \\
\vdots & \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
\text{Last round:} & \quad \cdots \quad \bullet
\end{align*}
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r, b)\). Analyze envy from \(r\) to \(b\).

\[
\begin{array}{cccc}
& b & r \\
First round: & \cdots & g & \cdots & \bullet & \cdots \\
Second round: & \cdots & \bullet & \cdots & \bullet & \cdots \\
. & . & . & . & . & . \\
Last round: & \cdots & \bullet & \cdots \\
\end{array}
\]
Round-Robin Algorithm

**Theorem:** For additive valuations, Round-Robin returns an EF1 allocation in polynomial time.

Fix a pair of agents \((r,b)\). Analyze envy from \(r\) to \(b\).

\[
\begin{align*}
\text{First round:} & \quad \cdots \quad g \quad \cdots \quad \bullet \quad \cdots \\
\text{Second round:} & \quad \cdots \quad \bullet \quad \cdots \quad \bullet \quad \cdots \\
\text{Last round:} & \quad \cdots \quad \bullet \quad \cdots
\end{align*}
\]

If \(b\) precedes \(r\), by additivity
\[
v_r(X_r) \geq v_r(X_b \setminus \{g\}).
\]
Definition: An allocation $X$ is envy free up to any item or EFX, if and only if for all agents $a_i, a_j$, and for all goods $g \in X_j$: $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

[Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]
**Definition:** An allocation \( X \) is envy free up to any item or EFX, if and only if for all agents \( a_i, a_j \), and for all goods \( g \in X_j \): 
\[ v_i(X_i) \geq v_i(X_j \setminus \{g\}) . \]

[Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]

Is the following allocation EFX?

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
<th>🍓</th>
<th>🍊</th>
<th>🍊</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Definition: An allocation $X$ is envy free up to any item or EFX, if and only if for all agents $a_i, a_j$, and for all goods $g \in X_j$: $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

[Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]

• EF $\implies$ EFX $\implies$ EF1
**Definition:** An allocation $X$ is **envy free up to any item** or EFX, if and only if for all agents $a_i, a_j$, and for all goods $g \in X_j$: $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

Do complete EFX allocations always exist?

[Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]
**EFX**

**Definition:** An allocation $X$ is **envy free up to any item** or **EFX**, if and only if for all agents $a_i, a_j$, and for all goods $g \in X_j$: $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

- $\text{EF} \implies \text{EFX} \implies \text{EF}^1$

Do complete EFX allocations always exist?  

[Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]
EFX

Definition: An allocation $X$ is envy free up to any item or EFX, if and only if for all agents $a_i, a_j$, and for all goods $g \in X_j$: $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

- $\text{EF} \implies \text{EFX} \implies \text{EF1}$

Do complete EFX allocations always exist? OPEN

Fair division’s biggest problem!

[Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]
**EFX**

**Definition:** An allocation $X$ is **envy free up to any item** or **EFX**, if and only if for all agents $a_i, a_j$, and for all goods $g \in X_j$: $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

- **EF $\implies$ EFX $\implies$ EF1**

Do complete EFX allocations always exist? **OPEN**

Fair division’s biggest problem!

In this seminar we will see:

- Complete EFX allocations exist for 3 agents if at least one has an additive valuation. [Akrami, Alon, Chaudhury, Garg, Mehlhorn, Mehta 2023]

- “Good” partial EFX allocations exists. [Chaudhury, Kavitha, Mehlhorn, Sgouritsa 2020]

[Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]
Fairness

- Envy Based
  - Envy Freeness
  - EF1
  - EFX

- Share Based
  - Proportionality
  - MMS
Proportionality

**Definition:** An allocation $X$ is proportional, if and only if for all agents $a_i$:

$$v_i(X_i) \geq v_i(M)/n.$$
**Proportionality**

**Definition:** An allocation $X$ is *proportional*, if and only if for all agents $a_i$: 
$v_i(X_i) \geq v_i(M)/n$.

Which allocation is proportional?
Proportionality

**Definition:** An allocation $X$ is **proportional**, if and only if for all agents $a_i$: $v_i(X_i) \geq v_i(M)/n$.

Do proportional allocations always exist?
Proportionality

Definition: An allocation $X$ is proportional, if and only if for all agents $a_i$: $v_i(X_i) \geq v_i(M)/n$.

Do proportional allocations always exist?

- For divisible goods, YES! (Next week)
Proportionality

Definition: An allocation $X$ is proportional, if and only if for all agents $a_i$:

$$v_i(X_i) \geq v_i(M)/n.$$ 

Do proportional allocations always exist?

- For divisible goods, YES! (Next week)
- For indivisible goods, NO!
**Definition:** An allocation $X$ is **proportional**, if and only if for all agents $a_i$:

$$v_i(X_i) \geq v_i(M)/n.$$

Do proportional allocations always exist?

- For divisible goods, YES! (Next week)
- For indivisible goods, NO!

I am not getting my proportional share!
Maximin Share

- What value can I guarantee for myself if I divide the items into $n$ bundles and receive the least valuable bundle?
Maximin Share

What value can I guarantee for myself if I divide the items into $n$ bundles and receive the least valuable bundle?

**Definition:** For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS_{v_i}^n(M) = \max_{(A_1, \ldots, A_n) \in [n]} \min_{j \in [n]} v_i(A_j).$$
Maximin Share

- What value can I guarantee for myself if I divide the items into \( n \) bundles and receive the least valuable bundle?

**Definition:** For all agents \( a_i \), maximin share of agent \( i \) is

\[
MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).
\]
Maximin Share

- What value can I guarantee for myself if I divide the items into $n$ bundles and receive the least valuable bundle?

**Definition:** For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).$$

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strawberry</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orange</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orange</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMS$_1$</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>Item</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>Item</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>Item</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
</tbody>
</table>

MMS$_1 = 3$
Maximin Share

- What value can I guarantee for myself if I divide the items into \( n \) bundles and receive the least valuable bundle?

**Definition:** For all agents \( a_i \), maximin share of agent \( i \) is

\[
MMS_i = MMS_{v_i}^{n} (M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).
\]

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Apple</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Strawberry</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\( MMS_1 = 3 \)

\( MMS_2 = 1 \)
Maximin Share

- What value can I guarantee for myself if I divide the items into $n$ bundles and receive the least valuable bundle?

**Definition:** For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).$$

<table>
<thead>
<tr>
<th></th>
<th>Banana</th>
<th>Apple</th>
<th>Strawberry</th>
<th>Orange</th>
<th>Mango</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>A2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

MMS$_1 = 3$

MMS$_2 = 1$

MMS$_3 = 2$
Maximin Share

**Definition:** For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n) \in [n]} \min_{j \in [n]} v_i(A_j).$$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>MMS$_1 = 3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>MMS$_2 = 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>MMS$_3 = 2$</td>
</tr>
</tbody>
</table>
Maximin Share

Definition: For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).$$

Definition: An allocation $X$ is MMS, if for all agents $a_i$, $v_i(X_i) \geq MMS_i$. 

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
<th>🍓</th>
<th>🍊</th>
<th>🍊</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Person 1]</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>![Person 2]</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>![Person 3]</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

MMS$_1 = 3$

MMS$_2 = 1$

MMS$_3 = 2$
Maximin Share

Definition: For all agents \( a_i \), maximin share of agent \( i \) is

\[
MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).
\]

Definition: An allocation \( X \) is MMS, if for all agents \( a_i \), \( v_i(X_i) \geq MMS_i \).

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>MMS_1 = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>MMS_2 = 1</td>
</tr>
<tr>
<td></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>MMS_3 = 2</td>
</tr>
<tr>
<td></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Maximin Share

**Definition:** For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).$$

**Definition:** An allocation $X$ is MMS, if for all agents $a_i$, $v_i(X_i) \geq MMS_i$. 
Maximin Share

Definition: For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n) \in [n]} \min_{j \in [n]} v_i(A_j).$$

Definition: An allocation $X$ is MMS, if for all agents $a_i$, $v_i(X_i) \geq MMS_i$.

Do MMS allocations always exist?
Maximin Share

**Definition:** For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).$$

**Definition:** An allocation $X$ is **MMS**, if for all agents $a_i$, $v_i(X_i) \geq MMS_i$.

Do MMS allocations always exist? **NO!** [Procaccia, Wang 2014]
Maximin Share

**Definition:** For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).$$

**Definition:** An allocation $X$ is **MMS**, if for all agents $a_i$, $v_i(X_i) \geq MMS_i$.

Do MMS allocations always exist? **NO!** [Procaccia, Wang 2014]

**Definition:** For all $\alpha \in [0, 1]$, an allocation $X$ is $\alpha$-MMS, if for all agents $a_i$, $v_i(X_i) \geq \alpha \cdot MMS_i$. 
Maximin Share

Definition: For all agents $a_i$, maximin share of agent $i$ is

$$MMS_i = MMS^n_{v_i}(M) = \max_{(A_1,\ldots,A_n)} \min_{j \in [n]} v_i(A_j).$$

Definition: An allocation $X$ is MMS, if for all agents $a_i$, $v_i(X_i) \geq MMS_i$.

Do MMS allocations always exist? NO! [Procaccia, Wang 2014]

Definition: For all $\alpha \in [0, 1]$, an allocation $X$ is $\alpha$-MMS, if for all agents $a_i$, $v_i(X_i) \geq \alpha \cdot MMS_i$.

- The best known $\alpha$: $3/4 + 3/3836$ [Akrami, Garg 2024]
Maximin Share

Definition: For all agents $a_i$, maximin share of agent $i$ is

$$
MMS_i = MMS^n_{v_i}(M) = \max_{(A_1, \ldots, A_n)} \min_{j \in [n]} v_i(A_j).
$$

Definition: An allocation $X$ is MMS, if for all agents $a_i$, $v_i(X_i) \geq MMS_i$.

Do MMS allocations always exist? NO! [Procaccia, Wang 2014]

Definition: For all $\alpha \in [0, 1]$, an allocation $X$ is $\alpha$-MMS, if for all agents $a_i$, $v_i(X_i) \geq \alpha \cdot MMS_i$.

- The best known $\alpha$: $3/4 + 3/3836$ [Akrami, Garg 2024]

In this seminar we will see:
- $3/4$-MMS allocations exist. [Ghodsi, Hajiaghayi, Seddighin, Seddighin, Yami 2018] [Garg, Taki 2020] [Akrami, Garg, Taki 2023]
Fairness

- Envy Based
  - Envy Freeness
  - EF1
  - EFX
- Share Based
  - Proportionality
  - MMS
Are we done?
Are we done?

<table>
<thead>
<tr>
<th></th>
<th>🍌</th>
<th>🍎</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Profile 1" /></td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td><img src="image2" alt="Profile 2" /></td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>
Are we done?

<table>
<thead>
<tr>
<th>100</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>🍌</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

Is the allocation “fair”?  
- EF1?
Are we done?

<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Is the allocation “fair”?  

- EF1?  
- EFX?
Are we done?

<table>
<thead>
<tr>
<th></th>
<th>Banana</th>
<th>Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

Is the allocation "fair"?

- EF1?
- EFX?
- MMS?
Efficiency

Divide *indivisible items* among *agents* in a *fair* and *efficient* manner.
Efficiency

Divide indivisible items among agents in a fair and efficient manner.

**Definition:** Allocation $X$ *pareto dominates* allocation $Y$, if and only if

- for all agents $a_i$, $v_i(X_i) \geq v_i(Y_i)$, and
- there exists an agent $a_j$, such that $v_j(X_j) > v_j(Y_j)$. 


Efficiency

Divide *indivisible items* among *agents* in a *fair* and *efficient* manner.

**Definition:** Allocation $X$ *pareto dominates* allocation $Y$, if and only if

- for all agents $a_i$, $v_i(X_i) \geq v_i(Y_i)$, and
- there exists an agent $a_j$, such that $v_j(X_j) > v_j(Y_j)$.

**Definition:** Allocation $X$ is *pareto optimal* or *PO* if there exists no allocation $Y$ such that $Y$ pareto dominates $X$. 
Efficiency

Divide **indivisible items** among **agents** in a **fair** and **efficient** manner.

**Definition:** Allocation $X$ **pareto dominates** allocation $Y$, if and only if
- for all agents $a_i$, $v_i(X_i) \geq v_i(Y_i)$, and
- there exists an agent $a_j$, such that $v_j(X_j) > v_j(Y_j)$.

**Definition:** Allocation $X$ is **pareto optimal** or PO if there exists no allocation $Y$ such that $Y$ pareto dominates $X$.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>![Banana]</td>
<td>![Apple]</td>
</tr>
<tr>
<td>![Person 1]</td>
<td>100</td>
</tr>
<tr>
<td>![Person 2]</td>
<td>1</td>
</tr>
</tbody>
</table>

Is the allocation pareto optimal?
Divide *indivisible items* among *agents* in a *fair* and *efficient* manner.

**Definition:** Allocation $X$ *pareto dominates* allocation $Y$, if and only if
- for all agents $a_i$, $v_i(X_i) \geq v_i(Y_i)$, and
- there exists an agent $a_j$, such that $v_j(X_j) > v_j(Y_j)$.

**Definition:** Allocation $X$ is *pareto optimal* or PO if there exists no allocation $Y$ such that $Y$ pareto dominates $X$.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="banana" /></td>
<td><img src="image" alt="apple" /></td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="user1" /></td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td><img src="image" alt="user2" /></td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

Is the allocation pareto optimal?
Efficiency

Divide **indivisible items** among **agents** in a **fair** and **efficient** manner.

**Definition:** Allocation $X$ **pareto dominates** allocation $Y$, if and only if
- for all agents $a_i$, $v_i(X_i) \geq v_i(Y_i)$, and
- there exists an agent $a_j$, such that $v_j(X_j) > v_j(Y_j)$.

**Definition:** Allocation $X$ is **pareto optimal** or **PO** if there exists no allocation $Y$ such that $Y$ pareto dominates $X$.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>Apple</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Is the allocation pareto optimal?</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
# Fairness and Efficiency

<table>
<thead>
<tr>
<th></th>
<th>Banana</th>
<th>Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

- **Fair**: ✔️
- **Efficient**: ✗️

<table>
<thead>
<tr>
<th></th>
<th>Banana</th>
<th>Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

- **Fair**: ✔️
- **Efficient**: ✔️

<table>
<thead>
<tr>
<th></th>
<th>Banana</th>
<th>Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

- **Fair**: ✔️
- **Efficient**: ✔️
Fairness and Efficiency

<table>
<thead>
<tr>
<th></th>
<th>Banana</th>
<th>Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

- **Fair**: A is fair, B is not fair.
- **Efficient**: A is efficient, B is efficient.

In this seminar we will see:

- EF1+PO allocations exist and can be computed in pseudopolynomial time.

[Barman, Krishnamurthy, Vaish 2018]
Nash Social Welfare

Definition: Nash social welfare of an allocation $X$ is

$$\text{NSW}(X) = \left( \prod_{a_i \in N} v_i(X_i) \right)^{1/n}.$$
Nash Social Welfare

Definition: Nash social welfare of an allocation $X$ is

$$\text{NSW}(X) = \left( \prod_{a_i \in N} v_i(X_i) \right)^{1/n}.$$ 

Definition: Allocation $X$ is MNW, if $\text{NSW}(X) \geq \text{NSW}(Y)$ for all allocations $Y$. 
Nash Social Welfare

Definition: Nash social welfare of an allocation $X$ is

$$NSW(X) = \left( \prod_{a_i \in N} v_i(X_i) \right)^{1/n}.$$ 

Definition: Allocation $X$ is $\alpha$-MNW, if $NSW(X) \geq \alpha \cdot NSW(Y)$ for all allocations $Y$ and $\alpha \in [0, 1]$. 
Nash Soical Welfare

Definition: Nash social welfare of an allocation $X$ is

$$\text{NSW}(X) = \left( \prod_{a_i \in N} v_i(X_i) \right)^{1/n}. $$

Definition: Allocation $X$ is $\alpha$-MNW, if $\text{NSW}(X) \geq \alpha \cdot \text{NSW}(Y)$ for all allocations $Y$ and $\alpha \in [0, 1]$.

In this seminar we will see:

- MNW $\implies$ EF1 + PO  [Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]
Nash Social Welfare

Definition: Nash social welfare of an allocation $X$ is

$$\text{NSW}(X) = \left( \prod_{a_i \in N} v_i(X_i) \right)^{1/n}.$$ 

Definition: Allocation $X$ is $\alpha$-MNW, if $\text{NSW}(X) \geq \alpha \cdot \text{NSW}(Y)$ for all allocations $Y$ and $\alpha \in [0, 1]$.

In this seminar we will see:

- MNW $\implies$ EF1 + PO  
  [Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016]

- $1.45^{-1}$-MNW allocations can be computed in polynomial time.  
  [Barman, Krishnamurthy, Vaish 2018]
Recap

Divide **items** among **agents** in a **fair** and **efficient** manner.

Notions of fairness: envy freeness, EF1, EFX, proportionality, MMS, . . .

Notions of efficiency: pareto optimality, MNW . . .
Seminar Overview

23.04: Introduction on Discrete Fair Division (HA)

30.04: Introduction on Cake Cutting (NR)

07.05: EFX: A Simpler Approach and an (Almost) Optimal Guarantee via Rainbow Cycle Number [Akrami, Alon, Chaudhury, Garg, Mehlhorn, Mehta 2023] (HA)
- EFX for 3 agents

14.05: Rental Harmony: Sperner’s Lemma in Fair Division [Su 1999] (NR)

21.05: no lecture

28.05: Fair and Efficient Cake Division with Connected Pieces [Arunachaleswaran, Barman, Kumar, Rathi 2019] (student talk)
-
Seminar Overview

04.06: The Unreasonable Fairness of Maximum Nash Welfare [Caragiannis, Kurokawa, Moulin, Procaccia, Shah, Wang 2016] (student talk)
- MNW $\implies$ EF1+PO

11.06: A Little Charity Guarantees Almost Envy-Freeness [Chaudhury, Kavitha, Mehlhorn, Sgouritsa 2020] (student talk)
- “good” partial EFX allocation

18.06: no lecture

25.06: Existence and Computation of Epistemic EFX Allocations [Caragiannis, Sharma, Garg, Rathi, Varricchio 2023] (student talk)
- a relaxation of EFX
Seminar Overview

02.07:  Simplification and Improvement of MMS Approximation [Akrami, Garg, Sharma, Taki 2023] (student talk)
- 3/4-MMS

09.07:  Finding Fair and Efficient Allocations [Barman, Krishnamurthy, Vaish 2018] (student talk)
- $1.45^{-1}$-MNW + EF1 + PO

16.07:  On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources [Bhaskar, Sricharan, Vaish 2021] (student talk)
-  

- randomized allocations
Don’t forget!

Send us your preferred list of the student papers by April 30th.