Fairness Criteria for Fair Division of Indivisible Goods

Sylvain Bouveret
LIG (STeamer) – Ensimag / Grenoble INP
Michel Lemaître
Formerly Onera Toulouse

Séminaire de l’équipe ROSP du laboratoire G-SCOP
Grenoble, March 24, 2016
You have:

- a finite set of objects $\mathcal{O} = \{1, \ldots, m\}$
- a finite set of agents $\mathcal{A} = \{1, \ldots, n\}$ having some preferences on the set of objects they may receive
Introduction

A fair division problem...

You have:
- a finite set of objects $\mathcal{O} = \{1, \ldots, m\}$
- a finite set of agents $\mathcal{A} = \{1, \ldots, n\}$ having some preferences on the set of objects they may receive

How would you allocate the objects to the agents so as to be as fair as possible?
You have:

- a finite set of **objects** \( O = \{1, \ldots, m\} \)
- a finite set of **agents** \( A = \{1, \ldots, n\} \) having some **preferences** on the set of objects they may receive

**How would you allocate the objects to the agents so as to be as fair as possible?**

More precisely, you want:

- an allocation \( \pi : A \rightarrow 2^O \)
- such that \( \pi_i \cap \pi_j = \emptyset \) if \( i \neq j \) (preemption),
- \( \bigcup_{i \in A} \pi_i = O \) (no free-disposal),
- and which takes into account the agents’ preferences
An ubiquitous problem

- Allocation of courses (or practical works) to students
- Allocation of take-off and landing slots in airports
- Allocation of tasks to workers
- Allocation of jobs to machines
- Allocation satellite resources
An ubiquitous problem

- Allocation of courses (or practical works) to students
- Allocation of take-off and landing slots in airports
- Allocation of tasks to workers
- Allocation of jobs to machines
- Allocation satellite resources
A rich problem
Introduction

A rich problem
A rich problem
A rich problem
Introduction

A rich problem
A rich problem

Introduction

Fair Division
Computational Social Choice (COMSOC)

COMSOC ≈ Social Choice ∩ Computer Science
Computational Social Choice (COMSOC)

COMSOC ≈ Social Choice ∩ Computer Science

1. Use techniques from economics to solve problems in IT (network sharing, job allocation...)

2. Use techniques from CS to analyze and solve economical problems (complexity of voting procedures, compact preference representation...)
Introduction

Computational social choice

Computational Social Choice (COMSOC)

COMSOC ≈ Social Choice ∩ Computer Science

1. Use techniques from economics to solve problems in IT (network sharing, job allocation...)

2. Use techniques from CS to analyze and solve economical problems (complexity of voting procedures, compact preference representation...)

- Fair division (indivisible goods or cake-cutting)
- Voting
- Coalition formation / hedonic games
- Judgment aggregation...
You have:

- a finite set of objects $\mathcal{O} = \{1, \ldots, m\}$
- a finite set of agents $\mathcal{A} = \{1, \ldots, n\}$ having some preferences on the set of objects they may receive
You have:

- a finite set of objects $\mathcal{O} = \{1, \ldots, m\}$
- a finite set of agents $\mathcal{A} = \{1, \ldots, n\}$ having some preferences on the set of objects they may receive

You want:

- an allocation $\pi : \mathcal{A} \rightarrow 2^\mathcal{O}$
- such that $\pi_i \cap \pi_j = \emptyset$ if $i \neq j$ (preemption),
- $\bigcup_{i \in \mathcal{A}} \pi_i = \mathcal{O}$ (no free-disposal),
- and which takes into account the agents’ preferences
A classical way to solve the problem:

- Ask each agent $i$ to give a score (weight, utility...) $w_i(o)$ to each object $o$.
- Consider all the agents have additive preferences:

$$u_i(\pi) = \sum_{o \in \pi} w_i(o)$$

- Find an allocation $\pi$ that:
The problem

Centralized allocation

A classical way to solve the problem:

- Ask each agent $i$ to give a score (weight, utility...) $w_i(o)$ to each object $o$
- Consider all the agents have additive preferences

$$
\rightarrow u_i(\pi) = \sum_{o \in \pi} w_i(o)
$$
- Find an allocation $\pi$ that:

1. maximizes the collective utility defined by a collective utility function,
   
   e.g. $uc(\pi) = \min_{i \in A} u_i(\pi_i) \rightarrow$ egalitarian solution
   [Bansal and Sviridenko, 2006]

2. or satisfies a given fairness criterion,
   
   e.g. $u_i(\pi_i) \geq u_i(\pi_j)$ for all agents $i, j$ – envy-freeness
   [Lipton et al., 2004].

---

The Santa Claus problem.
In Proceedings of STOC’06. ACM.

On approximately fair allocations of divisible goods.
In Proceedings of EC’04.
Example: 3 objects \(\{1, 2, 3\}\), 2 agents \(\{1, 2\}\).
Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>
Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

Egalitarian evaluation:

\[ \overrightarrow{\pi} = \langle \{1\}, \{2, 3\} \rangle \rightarrow uc(\overrightarrow{\pi}) = \min(5, 6 + 1) = 5 \]
Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

Egalitarian evaluation:

\(\pi = \langle\{1\}, \{2, 3\}\rangle \rightarrow uc(\pi) = \min(5, 6 + 1) = 5\)

\(\pi' = \langle\{1, 2\}, \{3\}\rangle \rightarrow uc(\pi') = \min(4 + 5, 6) = 6\)
Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

Egalitarian evaluation:
\(\pi = \langle\{1\}, \{2, 3\}\rangle\) \(\rightarrow uc(\pi) = \min(5, 6 + 1) = 5\)
\(\pi' = \langle\{1, 2\}, \{3\}\rangle\) \(\rightarrow uc(\pi') = \min(4 + 5, 6) = 6\)

Envy-freeness:
\(\pi\) is not envy-free (agent 1 envies agent 2)
**Example:** 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

**Preferences:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

**Egalitarian evaluation:**

\[ \pi = \langle \{1\}, \{2, 3\} \rangle \rightarrow uc(\pi) = \min(5, 6 + 1) = 5 \]

\[ \pi' = \langle \{1, 2\}, \{3\} \rangle \rightarrow uc(\pi') = \min(4 + 5, 6) = 6 \]

**Envy-freeness:**

\( \pi \) is **not** envy-free (agent 1 envies agent 2)

\( \pi' \) is envy-free.
In this work, we consider the 2\textsuperscript{nd} approach: choose a \textit{fairness property}, and find an allocation that satisfies it.
In this work, we consider the 2\textsuperscript{nd} approach: choose a \textit{fairness property}, and find an allocation that satisfies it.

\textbf{Problems:}

1. such an allocation does not always exist
   \[\rightarrow \text{\textit{e.g.} 2 agents, 1 object: no envy-free allocation exists}\]

2. many such allocations can exist
In this work, we consider the 2\textsuperscript{nd} approach: choose a \textit{fairness property}, and find an allocation that satisfies it.

\textbf{Problems:}

1. such an allocation does not always exist
   \[\Rightarrow\textit{e.g.} 2\text{ agents, 1 object: no envy-free allocation exists}\]

2. many such allocations can exist

\textbf{Idea:} consider several fairness properties, and try to satisfy the most demanding one.

In this work we consider five such properties.
Five fairness criteria

Envy-freeness
Envy-freeness

An allocation \( \pi \) is **envy-free** if no agent envies another one.

Formally: \( \forall i, j, u_i(\pi_i) \geq u_i(\pi_j) \)
**Envy-freeness**

An allocation $\vec{\pi}$ is **envy-free** if no agent envies another one.

Formally: $\forall i, j, u_i(\pi_i) \geq u_i(\pi_j)$

**Known facts:**

- An envy-free allocation may not exist.
- Deciding whether an allocation is envy-free is easy (quadratic time).
- Deciding whether an instance (agents, objects, preferences) has an envy-free allocation is hard – $\text{NP}$-complete [Lipton et al., 2004].

---

*Lipton, R., Markakis, E., Mossel, E., and Saberi, A. (2004).*

On approximately fair allocations of divisible goods.

In *Proceedings of EC’04.*
Five fairness criteria

Proportional fair share
Proportional fair share (PFS):

- Initially defined [Steinhaus, 1948] for continuous fair division (cake-cutting)

- **Idea:** each agent is “entitled” to at least the $n^{th}$ of the entire resource


The problem of fair division.

*Econometrica, 16*(1).
Proportional fair share (PFS):

- Initially defined [Steinhaus, 1948] for continuous fair division (cake-cutting)
- Idea: each agent is "entitled" to at least the $n^{th}$ of the entire resource

The problem of fair division.  
*Econometrica*, 16(1).

---

**Proportional fair share**

The **proportional fair share** of an agent $i$ is equal to:

$$u_i^{PFS} = \frac{u_i(\mathcal{O})}{n} = \sum_{o \in \mathcal{O}} \frac{w_i(o)}{n}$$

An allocation $\pi$ satisfies **(proportional) fair share** if every agent gets at least her fair share.
Easy or known facts:

- Deciding whether an allocation satisfies proportional fair share (PFS) is easy (linear time).
- For a given instance, there may be no allocation satisfying PFS → e.g. 2 agents, 1 object
- This is not true for cake-cutting (divisible resource) → Dubins-Spanier
Five fairness criteria

Proportional fair share: facts

Easy or known facts:
- Deciding whether an allocation satisfies proportional fair share (PFS) is easy (linear time).
- For a given instance, there may be no allocation satisfying PFS → e.g. 2 agents, 1 object
- This is not true for cake-cutting (divisible resource) → Dubins-Spanier

New (?) facts:
- Deciding whether an instance has an allocation satisfying PFS is hard even for 2 agents – NP-complete [PARTITION].
- $\pi'$ is envy-free $\Rightarrow \pi'$ satisfies PFS.$^1$

$^1$ Actually already noticed at least in an unpublished paper by Endriss, Maudet et al.
Proportional fair share: facts

Easy or known facts:

- Deciding whether an allocation satisfies proportional fair share (PFS) is easy (linear time).

- For a given instance, there may be no allocation satisfying PFS → e.g. 2 agents, 1 object

- This is not true for cake-cutting (divisible resource) → Dubins-Spanier

New (?) facts:

- Deciding whether an instance has an allocation satisfying PFS is hard even for 2 agents – NP-complete [PARTITION].

- \( \pi \) is envy-free ⇒ \( \pi \) satisfies PFS.\(^1\)

\(^1\) Actually already noticed at least in an unpublished paper by Endriss, Maudet et al.
Five fairness criteria

Max-min fair share
Five fairness criteria

Max-min fair share

PFS is nice, but sometimes too demanding for indivisible goods

→ e.g. 2 agents, 1 object
PFS is nice, but sometimes too demanding for indivisible goods → e.g. 2 agents, 1 object

Max-min fair share (MFS):

- Introduced recently [Budish, 2011]; not so much studied so far.
- **Idea**: in the cake-cutting case, PFS = the best share an agent can hopefully get for sure in a “I cut, you choose (I choose last)” game.
- Same game for indivisible goods → MFS.

The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes.
*Journal of Political Economy, 119*(*6*).
Max-min fair share

**Idea:** in the *cake-cutting* case, PFS = the best share an agent can hopefully get for sure in a "*I cut, you choose (I choose last)*" game.

**Max-min fair share**

The max-min fair share of an agent $i$ is equal to:

$$u_i^{\text{MFS}} \equiv \max_{\pi} \min_{j \in A} u_i(\pi_j)$$

An allocation $\pi$ satisfies max-min fair share (MFS) if every agent gets at least her max-min fair share.
Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>
Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

\[ u_{1}^{\text{MFS}} = 5 \text{ (with cut } \langle \{1\}, \{2, 3\} \rangle) \]
\[ u_{2}^{\text{MFS}} = 5 \text{ (with cut } \langle \{1, 2\}, \{3\} \rangle) \]
Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

\[ u_1^{\text{MFS}} = 5 \text{ (with cut } \langle\{1\}, \{2, 3\}\rangle) \]
\[ u_2^{\text{MFS}} = 5 \text{ (with cut } \langle\{1, 2\}, \{3\}\rangle) \]

MFS evaluation:
\[ \overrightarrow{\pi} = \langle\{1\}, \{2, 3\}\rangle \rightarrow u_1(\pi_1) = 5 \geq 5; u_2(\pi_2) = 7 \geq 5 \Rightarrow \text{MFS satisfied} \]
Example: 3 objects \( \{1, 2, 3\} \), 2 agents \( \{1, 2\} \).

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

\[ u_1^{\text{MFS}} = 5 \text{ (with cut } \langle \{1\}, \{2, 3\} \rangle \) \]
\[ u_2^{\text{MFS}} = 5 \text{ (with cut } \langle \{1, 2\}, \{3\} \rangle \) \]

MFS evaluation:

\[ \vec{\pi} = \langle \{1\}, \{2, 3\} \rangle \rightarrow u_1(\pi_1) = 5 \geq 5; u_2(\pi_2) = 7 \geq 5 \Rightarrow \text{MFS satisfied} \]
\[ \vec{\pi}'' = \langle \{2, 3\}, \{1\} \rangle \rightarrow u_1(\pi_1'') = 6 \geq 5; u_2(\pi_2'') = 4 < 5 \Rightarrow \text{MFS not satisfied} \]
Max-min fair share: examples

Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

\[
\begin{array}{ccc}
\text{agent 1} & 1 & 2 & 3 \\
5 & 4 & 2 \\
\text{agent 2} & 4 & 1 & 6 \\
\end{array}
\]

\[u_1^{\text{MFS}} = 5 \text{ (with cut } \langle\{1\}, \{2, 3\}\rangle)\]
\[u_2^{\text{MFS}} = 5 \text{ (with cut } \langle\{1, 2\}, \{3\}\rangle)\]

\text{MFS evaluation:}
\[
\overrightarrow{\pi} = \langle\{1\}, \{2, 3\}\rangle \rightarrow u_1(\pi_1) = 5 \geq 5; u_2(\pi_2) = 7 \geq 5 \Rightarrow \text{MFS satisfied}
\]
\[
\overrightarrow{\pi''} = \langle\{2, 3\}, \{1\}\rangle \rightarrow u_1(\pi_1') = 6 \geq 5; u_2(\pi_2') = 4 < 5 \Rightarrow \text{MFS not satisfied}
\]

Example: 2 agents, 1 object.
Max-min fair share: examples

Example: 3 objects \{1, 2, 3\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>agent 2</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

\[ u_1^{MFS} = 5 \text{ (with cut } \langle \{1\}, \{2, 3\} \rangle) \]
\[ u_2^{MFS} = 5 \text{ (with cut } \langle \{1, 2\}, \{3\} \rangle) \]

MFS evaluation:
\[ \overrightarrow{\pi} = \langle \{1\}, \{2, 3\} \rangle \rightarrow u_1(\pi_1) = 5 \geq 5; \quad u_2(\pi_2) = 7 \geq 5 \Rightarrow \text{MFS satisfied} \]
\[ \overrightarrow{\pi}'' = \langle \{2, 3\}, \{1\} \rangle \rightarrow u_1(\pi_1'') = 6 \geq 5; \quad u_2(\pi_2'') = 4 < 5 \Rightarrow \text{MFS not satisfied} \]

Example: 2 agents, 1 object.
\[ u_1^{MFS} = u_2^{MFS} = 0 \rightarrow \text{every allocation satisfies MFS!} \]

Not very satisfactory, but can we do much better?
Facts:

- Computing $u_i^{\text{MFS}}$ for a given agent is hard $\rightarrow$ NP-complete [\textsc{Partition}]
- Hence, deciding whether an allocation satisfies MFS is probably also hard (coNP-complete?)
- $\vec{\pi}$ satisfies PFS $\Rightarrow$ $\vec{\pi}$ satisfies MFS.
Facts:

- Computing $u_i^{MFS}$ for a given agent is hard $\rightarrow \text{NP-complete}$ [PARTITION]
- Hence, deciding whether an allocation satisfies MFS is probably also hard (coNP-complete?)
- $\overline{\pi}$ satisfies PFS $\Rightarrow$ $\overline{\pi}$ satisfies MFS.
Conjecture

For each instance there is at least one allocation satisfying max-min fair share.
Conjecture

For each instance there is at least one allocation satisfying max-min fair share.

- Proved for special cases (2 agents, matching,...), even very general ones (scoring functions...)
- No counterexample found on thousands of random instances.
Conjecture

For each instance there is at least one allocation satisfying max-min fair share.

- Proved for special cases (2 agents, matching,...), even very general ones (scoring functions...)
- No counterexample found on thousands of random instances.

The conjecture has been proved false by Procaccia and Wang using a very tricky counterexample (they also prove that 2/3 approximation is always achievable).

Fair enough: Guaranteeing approximate maximin shares.
forthcoming.
Max-min fair share: new facts

Since Procaccia and Wang’s work...
Max-min fair share: new facts

Since Procaccia and Wang’s work...

- Let $n \geq 3$:
  - if $m \leq n + 4$ an MMS allocation exists for sure [Kurokawa et al., 2015]
  - if $m \geq 3n + 4$ we can find an instance without MMS allocation [Kurokawa et al., 2015]
  - in between?

- 2/3-approximation in Polynomial time [Amanatidis et al., 2015] (7/8 for the 3-agent case)

- An MMS allocation exists with (theoretical) very high probability [Amanatidis et al., 2015, Kurokawa et al., 2015]

---

Approximation algorithms for computing maximin share allocations.
In ICALP (1), volume 9134 of Lecture Notes in Computer Science, pages 39–51. Springer.

When can the maximin share guarantee be guaranteed?
“Max-min fair share” sounds like “max-min optimality”...

**Idea:** Use the egalitarian approach to compute

$$\hat{\pi} = \arg\max_{\pi} (\min_{i \in A} u_i(\pi_i))$$

Santa-Claus problem [Bansal and Sviridenko, 2006] (connection to maximum makespan minimization in job scheduling on multiple machines), and it will give an MFS allocation

---

*Bansal, N. and Sviridenko, M. (2006).*

The Santa Claus problem.  
*In Proceedings of STOC’06. ACM.*
“Max-min fair share” sounds like “max-min optimality”...

**Idea:** Use the egalitarian approach to compute

\[
\hat{\pi} = \arg\max_{\pi} \left( \min_{i \in A} u_i(\pi_i) \right)
\]

Santa-Claus problem [Bansal and Sviridenko, 2006] (connection to maximum makespan minimization in job scheduling on multiple machines), and it will give an MFS allocation

**Bad luck:** there exist instances with MMS allocations, for which
{MMS allocations \(\cap\) (lexi-)min optimal allocations} = \(\emptyset\).

---

*Bansal, N. and Sviridenko, M. (2006).*

The Santa Claus problem.

In *Proceedings of STOC’06*. ACM.
Five fairness criteria

Computing a MFS allocation

- 2/3-approximation in Polynomial time [Amanatidis et al., 2015] (7/8 for the 3-agent case)
- **Open question**: complexity of deciding whether an instance is MFS?
- **Open question**: computing an MFS allocation (when there is one...) efficiently (Santa-Claus may help but is not the answer)

Approximation algorithms for computing maxmin share allocations.
Five fairness criteria

Min-max fair share
Min-max fair share

- Max-min fair share: “I cut, you choose (I choose last)”
Five fairness criteria

Min-max fair share

- Max-min fair share: “I cut, you choose (I choose last)”
- Idea: why not do the opposite (“Someone cuts, I choose first”)?

\[
\text{Min-max fair share (mFS)}
\]

The min-max fair share of an agent \(i\) is equal to:

\[
u_{\text{mFS}}(i) = \min_{\pi \in \mathcal{A}} \max_{j \in A} u_i(\pi_j)
\]

An allocation \(\pi\) satisfies min-max fair share (mFS) if every agent gets at least her min-max fair share. mFS = the worst share an agent can get in a “Someone cuts, I choose first” game. In the cake-cutting case, same as PFS.
Min-max fair share

- Max-min fair share: “I cut, you choose (I choose last)”
- Idea: why not do the opposite (“Someone cuts, I choose first”)? → Min-max fair share

Min-max fair share (mFS)

The min-max fair share of an agent $i$ is equal to:

$$u_i^{mFS} \overset{\text{def}}{=} \min_{\pi} \max_{j \in A} u_i(\pi_j)$$

An allocation $\pi$ satisfies min-max fair share (mFS) if every agent gets at least her min-max fair share.
Min-max fair share

- Max-min fair share: “I cut, you choose (I choose last)"
- Idea: why not do the opposite (“Someone cuts, I choose first”)?

→ Min-max fair share

Min-max fair share (mFS)

The min-max fair share of an agent \( i \) is equal to:

\[
  u_i^{\text{mFS}} \overset{\text{def}}{=} \min_{\overrightarrow{\pi}} \max_{j \in A} u_i(\pi_j)
\]

An allocation \( \overrightarrow{\pi} \) satisfies min-max fair share (mFS) if every agent gets at least her min-max fair share.

- mFS = the worst share an agent can get in a “Someone cuts, I choose first” game.
- In the cake-cutting case, same as PFS.
Facts:

- Computing $u_{i}^{mFS}$ for a given agent is hard $\rightarrow \text{coNP-complete}$ [\text{PARTITION}]

- Hence, deciding whether an allocation satisfies mFS is probably also hard (\text{NP-complete}?).

- $\overrightarrow{\pi}$ satisfies mFS $\Rightarrow$ $\overrightarrow{\pi}$ satisfies PFS.

- $\overrightarrow{\pi}$ is envy-free $\Rightarrow$ $\overrightarrow{\pi}$ satisfies mFS.
**Facts:**

- Computing $u_i^{\text{mFS}}$ for a given agent is hard $\rightarrow \text{coNP}$-complete [PARTITION].
- Hence, deciding whether an allocation satisfies mFS is probably also hard (NP-complete?).
- $\pi$ satisfies mFS $\Rightarrow$ $\pi$ satisfies PFS.
- $\pi$ is envy-free $\Rightarrow$ $\pi$ satisfies mFS.
Five fairness criteria

*Competitive Equilibrium from Equal Incomes*
Competitive Equilibrium from Equal Incomes.

Example: 4 objects \( \{1, 2, 3, 4\} \), 2 agents \( \{1, 2\} \).
**Competitive Equilibrium from Equal Incomes.**

*Example:* 4 objects \{1, 2, 3, 4\}, 2 agents \{1, 2\}.

**Preferences:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>7</td>
<td>2</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>agent 2</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>
Competitive Equilibrium from Equal Incomes.

Example: 4 objects \(\{1, 2, 3, 4\}\), 2 agents \(\{1, 2\}\).

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>7</td>
<td>2</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>agent 2</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

For €1, what would you buy?
Competitive Equilibrium from Equal Incomes.

Example: 4 objects \{1, 2, 3, 4\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>7</td>
<td>2</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>agent 2</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

For €1, what would you buy?

- Agent 1: 1 and 4;
- Agent 2: 2 and 3.

Disjoint shares!
Competitive Equilibrium from Equal Incomes.

Example: 4 objects \{1, 2, 3, 4\}, 2 agents \{1, 2\}.

Preferences:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>7</td>
<td>2</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>agent 2</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

For €1, what would you buy?

- Agent 1: 1 and 4;
- Agent 2: 2 and 3.

⇒ Disjoint shares!

Allocation \(\{\{1, 4\}, \{2, 3\}\}\), with prices \(\langle 0.8, 0.2, 0.8, 0.2 \rangle\) forms a CEEI.
⇒ Allocation \(\{\{1, 4\}, \{2, 3\}\}\) satisfies CEEI.
Five fairness criteria

Competitive Equilibrium from Equal Incomes

- A classical notion in economics [Moulin, 1995]
- Subcase (indivisible goods) of the Fisher model [Walras, 1874, Fisher, 1892]
- Introduced recently in computer science [Othman et al., 2010]

Fisher, I. (1892).
*Mathematical Investigations in the Theory of Value and Prices, and Appreciation and Interest.*
Augustus M. Kelley, Publishers.

*Cooperative Microeconomics, A Game-Theoretic Introduction.*
Prentice Hall.

Finding approximate competitive equilibria: efficient and fair course allocation.
In *Proceedings of AAMAS’10.*

Walras, L. (1874).
*Éléments d’économie politique pure ou Théorie de la richesse sociale.*
L. Corbaz, 1 edition.
Fact: $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.
Fact: $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.
Fact: $\overrightarrow{\pi}$ satisfies CEEI $\Rightarrow$ $\overrightarrow{\pi}$ is envy-free.

Fisher model: an equilibrium always exists – Nash (×) optimal
**Fact:** $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.

![Diagram showing fairness criteria hierarchy]

**Fisher model:** an equilibrium always exists – Nash ($\times$) optimal

Unfortunately, in the **discrete setting**, a CEEI may not exist.
Five fairness criteria

**CEEI: known facts**

**Fact:** \( \pi \) satisfies CEEI \( \Rightarrow \) \( \pi \) is envy-free.

\[
\begin{align*}
\text{max-min fair share} & \quad \downarrow \quad \text{proportional fair share} & \quad \downarrow \quad \text{min-max fair share} \\
\text{(weaker)} & \quad \text{(CEEI)} & \quad \text{(stronger)} \\
\text{envy-freeness} & \quad \downarrow \quad \text{enjoy-freeness}
\end{align*}
\]

**Fisher model:** an equilibrium always exists – Nash (\( \times \)) optimal

Unfortunately, in the **discrete setting**, a CEEI may not exist.

Worse [Brânzei et al., 2015]...

- Is \( (\pi, \pi) \) a CEEI? \( \rightarrow \) coNP-complete
- Does there exist a CEEI? NP-hard

---

Characterization and computation of equilibria for indivisible goods.
In *Algorithmic Game Theory*, pages 244–255. Springer.
Five fairness criteria

CEEI: open problems

- Does there exist a CEEI? $\text{NP}$-hard and in $\Sigma_2^P$. Precise complexity?
- How to test whether $\vec{\pi}$ is a CEEI (and find the associated $\vec{p}$)?
Five fairness criteria

CEEI: open problems

- Does there exist a CEEI? NP-hard and in $\Sigma_2^P$. Precise complexity?
- How to test whether $\vec{\pi}$ is a CEEI (and find the associated $\vec{\rho}$)?

\[
0 \leq p_o \leq 1, \text{ for all } o \in [1, m] \tag{1}
\]
\[
\sum_{o=1}^{m} a_{\pi_i, o} p_o \leq 1, \text{ for all } i \in [1, n], \text{ with } a_{\pi_i, o} = 1 \text{ if } o \in \pi_i, 0 \text{ otherwise} \tag{2}
\]
\[
\sum_{o=1}^{m} a_{\pi', o} p_o > 1, \text{ for all } \pi' \text{ such that } \exists i \text{ such that } u_i(\pi') > u_i(\pi_i) \tag{3}
\]
Five fairness criteria

**CEEI: open problems**

- Does there exist a CEEI? \textbf{NP}-hard and in $\Sigma^P_2$. Precise complexity?
- How to test whether $\pi$ is a CEEI (and find the associated $\vec{p}$)?

\begin{align*}
0 \leq p'_o, & \text{ for all } o \in [1, m] \quad (1) \\
\sum_{o=1}^{m} a_{\pi_i, o} p'_o \leq d, & \text{ for all } i \in [1, n], \text{ with } a_{\pi_i, o} = 1 \text{ if } o \in \pi_i, 0 \text{ otherwise} \quad (2) \\
\sum_{o=1}^{m} a_{\pi', o} p_o \geq d + 1, & \text{ for all } \pi' \text{ such that } \exists i \text{ such that } u_i(\pi') > u_i(\pi_i) \quad (3)
\end{align*}
Five fairness criteria

CEEI: open problems

- Does there exist a CEEI? \textbf{NP}-hard and in $\Sigma_2^P$. Precise complexity?
- How to test whether $\pi$ is a CEEI (and find the associated $\overline{p}$)?

\begin{align*}
0 \leq p'_o, & \text{ for all } o \in [1, m] \quad (1) \\
\sum_{o=1}^{m} a_{\pi_i, o} p'_o \leq d, & \text{ for all } i \in [1, n], \text{ with } a_{\pi_i, o} = 1 \text{ if } o \in \pi_i, 0 \text{ otherwise} \quad (2) \\
\sum_{o=1}^{m} a_{\pi', o} p_o \geq d + 1, & \text{ for all } \pi' \text{ such that } \exists i \text{ such that } u_i(\pi') > u_i(\pi_i) \quad (3)
\end{align*}

- How to compute a CEEI allocation?
Five fairness criteria

**CEEI: open problems**

- Does there exist a CEEI? **NP**-hard and in $\Sigma_2^P$. Precise complexity?
- How to test whether $\pi$ is a CEEI (and find the associated $\vec{p}$)?

\[0 \leq p'_o, \text{ for all } o \in [1, m]\]  
(1)

\[\sum_{o=1}^{m} a_{\pi_i,o}p'_o \leq d, \text{ for all } i \in [1, n], \text{ with } a_{\pi_i,o} = 1 \text{ if } o \in \pi_i, \text{ 0 otherwise}\]  
(2)

\[\sum_{o=1}^{m} a_{\pi',o}p_o \geq d + 1, \text{ for all } \pi' \text{ such that } \exists i \text{ such that } u_i(\pi') > u_i(\pi_i)\]  
(3)

- How to compute a CEEI allocation?

**Simplistic algorithm:** compute all allocations and test which ones are CEEI.
Five fairness criteria

Summary and interpretation
Five fairness criteria

Interpretation

- max-min fair share
- proportional fair share
- min-max fair share
- envy-freeness
- CEEI

For all allocation $\pi$:

$$\leftarrow \pi \models \text{CEEI} \Rightarrow \pi \models \text{EF} \Rightarrow \pi \models \text{mFS} \Rightarrow \pi \models \text{PFS} \Rightarrow \pi \models \text{MFS} \rightarrow$$

the highest property $\pi$ satisfies, the most satisfactory it is.

If $I|P$ is the set of instances s.t at least one allocation satisfies $P$:

$$I|\text{CEEI} \subset I|\text{EF} \subset I|\text{mFS} \subset I|\text{PFS} \subset I|\text{MFS} \rightarrow$$

the lowest subset, the less "conflict-prone".
Five fairness criteria

Interpretation

<table>
<thead>
<tr>
<th>max-min fair share</th>
<th>proportional fair share</th>
<th>min-max fair share</th>
<th>CEEI</th>
</tr>
</thead>
<tbody>
<tr>
<td>weaker</td>
<td></td>
<td>stronger</td>
<td></td>
</tr>
</tbody>
</table>

1. For all allocation $\pi$:

$$(\pi \models \text{CEEI}) \Rightarrow (\pi \models \text{EF}) \Rightarrow (\pi \models \text{mFS}) \Rightarrow (\pi \models \text{PFS}) \Rightarrow (\pi \models \text{MFS})$$

→ the highest property $\pi$ satisfies, the most satisfactory it is.
For all allocation $\vec{\pi}$:

\[(\vec{\pi} \models \text{CEEI}) \Rightarrow (\vec{\pi} \models \text{EF}) \Rightarrow (\vec{\pi} \models \text{mFS}) \Rightarrow (\vec{\pi} \models \text{PFS}) \Rightarrow (\vec{\pi} \models \text{MFS})\]

→ the highest property $\vec{\pi}$ satisfies, the most satisfactory it is.

If $\mathcal{I}|_{\mathcal{P}}$ is the set of instances s.t at least one allocation satisfies $\mathcal{P}$:

\[\mathcal{I}|_{\text{CEEI}} \subset \mathcal{I}|_{\text{EF}} \subset \mathcal{I}|_{\text{mFS}} \subset \mathcal{I}|_{\text{PFS}} \subset \mathcal{I}|_{\text{MFS}} \subset \mathcal{I}\]

→ the lowest subset, the less “conflict-prone”.

1. max-min fair share
2. proportional fair share
3. min-max fair share
4. envy-freeness
5. CEEI

weaker

stronger
Five fairness criteria

Interpretation

max-min fair share \downarrow \text{ proportional fair share} \downarrow \text{ envy-freeness} \downarrow \text{ CEEI}

weaker \text{ stronger}

1. For all allocation $\pi$:

$$\pi \models \text{CEEI} \Rightarrow \pi \models \text{EF} \Rightarrow \pi \models \text{mFS} \Rightarrow \pi \models \text{PFS} \Rightarrow \pi \models \text{MFS}$$

→ the highest property $\pi$ satisfies, the most satisfactory it is.

2. If $\mathcal{I}_P$ is the set of instances s.t at least one allocation satisfies $P$:

$$\mathcal{I}_\text{CEEI} \subset \mathcal{I}_\text{EF} \subset \mathcal{I}_\text{mFS} \subset \mathcal{I}_\text{PFS} \subset \mathcal{I}_\text{MFS} \subset \mathcal{I}$$

→ the lowest subset, the less “conflict-prone”.

Two extreme examples:

- 2 agents, 1 object → only in $\mathcal{I}_\text{MFS}$
- 2 agents, 2 objects, with

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent 1</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>agent 2</td>
<td>0</td>
<td>1000</td>
</tr>
</tbody>
</table>

→ in $\mathcal{I}_\text{CEEI}$ (with e.g. $\vec{p} = (1, 1)$).
A glimpse at experiments

What about fairness criteria in practice?
What about fairness criteria in practice?

**Goal of our experiments:** evaluate the distribution of the allocation over:
- the fairness scale (–, MFS, PFS, mFS, EF, CEEI);
- three efficiency levels (–, Sequenceable, Pareto-efficient).
A glimpse at experiments

100 random instances (3 agents, 10 objects)
Experiments

A glimpse at experiments

100 random instances (3 agents, 10 objects)
Additive preferences are nice but have a limited expressiveness.
Additive preferences are nice but have a limited expressiveness.

**Examples:**
- the pair of skis and the pair of ski poles (complementarity)
- the pair of skis and the snowboard (substitutability)
Additive preferences are nice but have a limited expressiveness.

**Examples:**
- the pair of skis and the pair of ski poles (complementarity)
  \[
  u(\{\text{skis}, \text{poles}\}) > u(\text{skis}) + u(\text{poles})
  \]
- the pair of skis and the snowboard (substitutability)
- Additive preferences are nice but have a limited expressiveness.

- Examples:
  - the pair of skis and the pair of ski poles (complementarity)
    \[ u(\{\text{skis}, \text{poles}\}) > u(\text{skis}) + u(\text{poles}) \]
  - the pair of skis and the snowboard (substitutability)
    \[ u(\{\text{skis}, \text{snowboard}\}) < u(\text{skis}) + u(\text{snowboard}) \]
A glimpse beyond additive preferences

k-additive preferences

- Additive preferences are nice but have a limited expressiveness.
- Examples:
  - the pair of skis and the pair of ski poles (complementarity)
    \[ u(\{\text{skis, poles}\}) > u(\text{skis}) + u(\text{poles}) \]
  - the pair of skis and the snowboard (substitutability)
    \[ u(\{\text{skis, snowboard}\}) < u(\text{skis}) + u(\text{snowboard}) \]

k-additive preferences

A weight \( w(S) \) to each subset \( S \) of objects (not only singletons) of size \( \leq k \).

Note: additive = 1-additive
$k$-additive preferences

- Additive preferences are nice but have a limited expressiveness.
- **Examples:**
  - the pair of skis and the pair of ski poles (complementarity)
    \[ u(\{\text{skis}, \text{poles}\}) > u(\text{skis}) + u(\text{poles}) \]
  - the pair of skis and the snowboard (substitutability)
    \[ u(\{\text{skis}, \text{snowboard}\}) < u(\text{skis}) + u(\text{snowboard}) \]

$k$-additive preferences

A weight $w(S)$ to each subset $S$ of objects (not only singletons) of size $\leq k$.
**Note:** additive = 1-additive

**Examples:**

- $w(\text{skis}) = 10$; $w(\text{poles}) = 0$; $w(\{\text{skis, poles}\}) = 90$
  \[ u(\{\text{skis, poles}\}) = 100 > 10 + 0 \]

- $w(\text{skis}) = 100$; $w(\text{snowboard}) = 100$; $w(\{\text{skis, snowboard}\}) = -100$
  \[ u(\{\text{skis, snowboard}\}) = 100 < 100 + 100 \]
Reminder

For **additive preferences** we can almost always find an allocation satisfying max-min fair share.
Reminder

For **additive preferences** we can almost always find an allocation satisfying max-min fair share.

For **$k$-additive preferences** ($k \geq 2$) this is obviously not true:

**Example:** 4 objects, 2 agents

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>×</td>
<td></td>
<td>×</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>×</td>
<td></td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>
Reminder

For additive preferences we can almost always find an allocation satisfying max-min fair share.

For $k$-additive preferences ($k \geq 2$) this is obviously not true:

Example: 4 objects, 2 agents

Agent 1: $w(\{1, 2\}) = w(\{3, 4\}) = 1 \rightarrow u_1^{\text{MFS}} = 1$
Reminder

For additive preferences we can almost always find an allocation satisfying max-min fair share.

For \( k \)-additive preferences \( (k \geq 2) \) this is obviously not true:

**Example:** 4 objects, 2 agents

\[
\begin{align*}
\text{Agent 1: } & \ w(\{1, 2\}) = w(\{3, 4\}) = 1 \rightarrow u_1^{\text{MFS}} = 1 \\
\text{Agent 2: } & \ w(\{1, 4\}) = w(\{2, 3\}) = 1 \rightarrow u_2^{\text{MFS}} = 1
\end{align*}
\]
A glimpse beyond additive preferences

MFS and \(k\)-additive preferences

Reminder

For additive preferences we can almost always find an allocation satisfying max-min fair share.

For \(k\)-additive preferences \((k \geq 2)\) this is obviously not true:

Example: 4 objects, 2 agents

Agent 1: \(w(\{1, 2\}) = w(\{3, 4\}) = 1 \rightarrow u_1^{MFS} = 1\)

Agent 2: \(w(\{1, 4\}) = w(\{2, 3\}) = 1 \rightarrow u_2^{MFS} = 1\)

Worse... Deciding whether there exists one is NP-complete [\textsc{Partition}].
A scale of properties (for numerical additive preferences)...
A scale of properties (for numerical additive preferences)...

Max-min fair share
Almost always possible to satisfy it
A scale of properties (for numerical additive preferences)...

**Proportional fair share**
Cannot be satisfied *e.g.* in the 1 object, 2 agents case

**Max-min fair share**
Almost always possible to satisfy it
A scale of properties (for numerical additive preferences)...

- **Min-max fair share**
- **Proportional fair share**
  Cannot be satisfied *e.g.* in the 1 object, 2 agents case
- **Max-min fair share**
  Almost always possible to satisfy it
A scale of properties (for numerical additive preferences)...

- **Envy-freeness**
  Requires somewhat complementary preferences

- **Min-max fair share**

- **Proportional fair share**
  Cannot be satisfied *e.g.* in the 1 object, 2 agents case

- **Max-min fair share**
  Almost always possible to satisfy it
A scale of properties (for numerical additive preferences)...

- **Competitive Equilibrium from Equal Incomes**
  Requires complementary preferences

- **Envy-freeness**
  Requires somewhat complementary preferences

- **Min-max fair share**

- **Proportional fair share**
  Cannot be satisfied e.g. in the 1 object, 2 agents case

- **Max-min fair share**
  Almost always possible to satisfy it
A scale of properties (for numerical additive preferences)...

- **Competitive Equilibrium from Equal Incomes**
  Requires complementary preferences

- **Envy-freeness**
  Requires somewhat complementary preferences

- **Min-max fair share**

- **Proportional fair share**
  Cannot be satisfied *e.g.* in the 1 object, 2 agents case

- **Max-min fair share**
  Almost always possible to satisfy it

A possible approach to fairness in multiagent resource allocation problems:

1. Determine the highest satisfiable criterion.
2. Find an allocation that satisfies this criterion.
3. Explain to the upset agents that we cannot do much better.
Conclusion

What else

Some future directions...

- Link with a scale of efficiency criteria (recent work)
- Some missing complexity results
- Develop efficient algorithms
- More experiments
- Extend to more expressive preference languages (including ordinal ones...)

Fairness Criteria for Fair Division of Indivisible Goods