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

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

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?*

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?*

More precisely, you want:

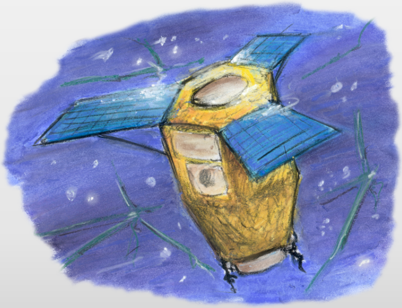
- an allocation $\overrightarrow{\pi} : \mathcal{A} \to 2^{\mathcal{O}}$
- such that $\pi_i \cap \pi_j = \emptyset$ if $i \neq j$ (preemption),
- $\bigcup_{i \in \mathcal{A}} \pi_i = \mathcal{O}$ (no free-disposal),
- and which takes into account the agents' preferences

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

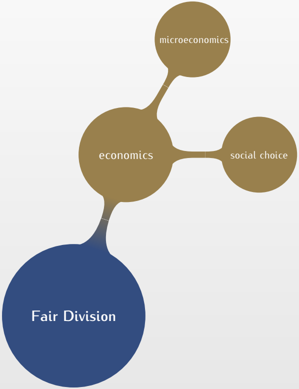
- Allocation of courses (or practical works) to students
- Allocation of take-off and landing slots in airports
- Allocation of tasks to workers
- Allocation of jobs to machines
- Allocation satellite resources



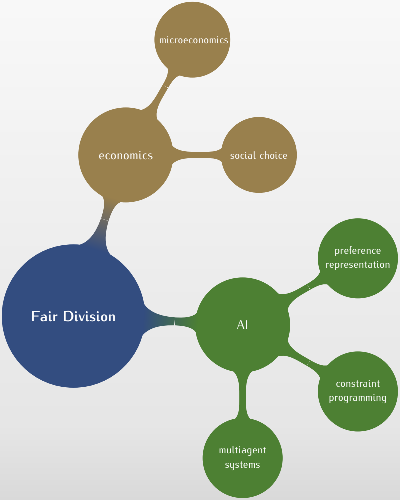
A rich problem

Fair Division

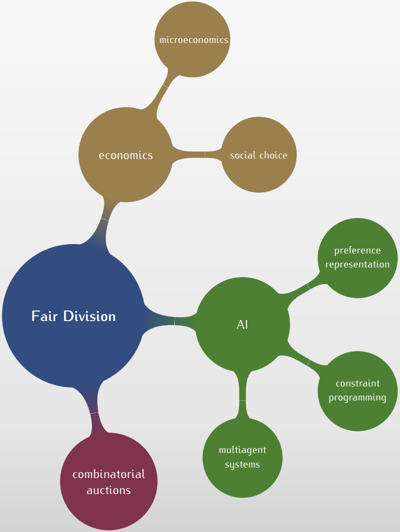
A rich problem



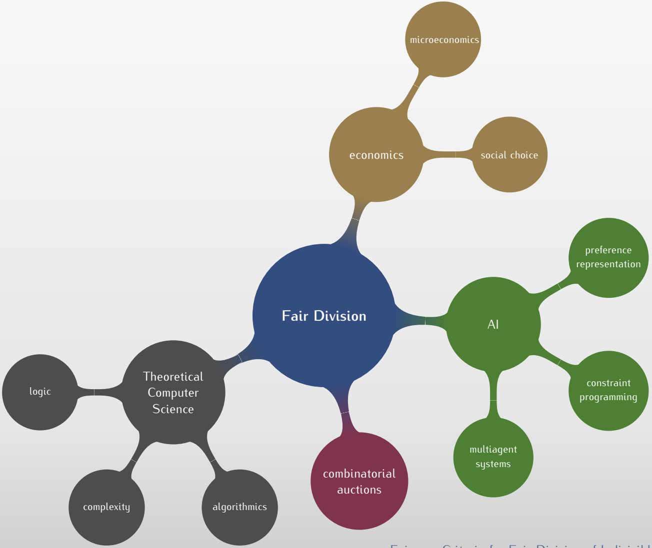
A rich problem



A rich problem



A rich problem



A rich problem

Computational social choice

Computational Social Choice (COMSOC)

COMSOC $\approx$ Social Choice $\cap$ Computer Science

Computational social choice

Computational Social Choice (COMSOC)

COMSOC $\approx$ Social Choice $\cap$ 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...)

Computational social choice

Computational Social Choice (COMSOC)

COMSOC $\approx$ Social Choice $\cap$ 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...

Back to our 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

Back to our 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

You want:

- an allocation $\overrightarrow{\pi} : \mathcal{A} \to 2^{\mathcal{O}}$
- 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

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 $\overrightarrow{\pi}$ that:

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 $\overrightarrow{\pi}$ that:

1 | maximizes the collective utility defined by a **collective utility function**,

e.g. $uc(\overrightarrow{\pi}) = \min_{i \in \mathcal{A}} u_i(\pi_i)$ – 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].

Bansal, N. and Sviridenko, M. (2006).
The Santa Claus problem.
In *Proceedings of STOC'06*. ACM.

Lipton, R., Markakis, E., Mossel, E., and Saberi, A. (2004).
On approximately fair allocations of divisible goods.
In *Proceedings of EC'04*.

Example

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

Example

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

Example

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

Egalitarian evaluation:

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

Example

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

Egalitarian evaluation:

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

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

Example

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

Egalitarian evaluation:

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

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

Envy-freeness:

$\overrightarrow{\pi}$ is **not** envy-free (agent 1 envies agent 2)

Example

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

Egalitarian evaluation:

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

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

Envy-freeness:

$\overrightarrow{\pi}$ is **not** envy-free (agent 1 envies agent 2)

$\overrightarrow{\pi}'$ is envy-free.

Fairness properties

In this work, we consider the 2^{nd} approach: choose a **fairness property**, and find an allocation that satisfies it.

Fairness properties

In this work, we consider the 2^{nd} approach: choose a **fairness property**, and find an allocation that satisfies it.

Problems:

1 such an allocation does not always exist
$\rightarrow$ *e.g.* 2 agents, 1 object: no envy-free allocation exists

2 many such allocations can exist

Fairness properties

In this work, we consider the 2nd approach: choose a **fairness property**, and find an allocation that satisfies it.

Problems:

1. such an allocation does not always exist
   → *e.g.* 2 agents, 1 object: no envy-free allocation exists
2. many such allocations can exist

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

Envy-freeness

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

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

Envy-freeness

Envy-freeness

An allocation $\overrightarrow{\pi}$ 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 – **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

Proportional fair share (PFS):

- Initially defined [Steinhaus, 1948] for continuous fair division (*cake-cutting*)
- **Idea:** each agent is "entitled" to at least the nth of the entire resource

Steinhaus, H. (1948).
The problem of fair division.
Econometrica, 16(1).

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

Steinhaus, H. (1948).
The problem of fair division.
Econometrica, 16(1).

Proportional fair share

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

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

An allocation $\overrightarrow{\pi}$ satisfies **(proportional) fair share** if every agent gets at least her fair share.

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
  $\rightarrow$ *e.g.* 2 agents, 1 object
- This is not true for cake-cutting (divisible resource)
  $\rightarrow$ Dubins-Spanier

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
  $\rightarrow$ *e.g.* 2 agents, 1 object
- This is not true for cake-cutting (divisible resource)
  $\rightarrow$ Dubins-Spanier

New (?) facts:

- Deciding whether an instance has an allocation satisfying PFS is hard even for 2 agents – **NP**-complete [PARTITION].
- $\vec{\pi}$ is envy-free $\Rightarrow$ $\vec{\pi}$ 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
  $\rightarrow$ *e.g.* 2 agents, 1 object
- This is not true for cake-cutting (divisible resource)
  $\rightarrow$ Dubins-Spanier

New (?) facts:

- Deciding whether an instance has an allocation satisfying PFS is hard even for 2 agents – **NP**-complete [PARTITION].
- $\vec{\pi}$ is envy-free $\Rightarrow$ $\vec{\pi}$ 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

Max-min fair share

PFS is nice, but sometimes too demanding for indivisible goods

$\rightarrow$ *e.g.* 2 agents, 1 object

Max-min fair share

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.

Budish, E. (2011).
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:

$$u_i^{\text{MFS}} \stackrel{def}{=} \max_{\overrightarrow{\pi}} \min_{j \in \mathcal{A}} u_i(\pi_j)$$

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

Max-min fair share: examples

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

Max-min fair share: examples

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

$\rightarrow u_1^{\text{MFS}} = 5$ (with cut $\langle\{1\}, \{2, 3\}\rangle$)

$\rightarrow u_2^{\text{MFS}} = 5$ (with cut $\langle\{1, 2\}, \{3\}\rangle$)

Max-min fair share: examples

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

$\rightarrow u_1^{\text{MFS}} = 5$ (with cut $\langle\{1\},\{2,3\}\rangle$)

$\rightarrow u_2^{\text{MFS}} = 5$ (with cut $\langle\{1,2\},\{3\}\rangle$)

MFS evaluation:

$\vec{\pi} = \langle\{1\},\{2,3\}\rangle \rightarrow u_1(\pi_1) = 5 \geq 5$; $u_2(\pi_2) = 7 \geq 5 \Rightarrow$ MFS satisfied

Max-min fair share: examples

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

$\rightarrow u_1^{\text{MFS}} = 5$ (with cut $\langle\{1\},\{2,3\}\rangle$)

$\rightarrow u_2^{\text{MFS}} = 5$ (with cut $\langle\{1,2\},\{3\}\rangle$)

MFS evaluation:

$\overrightarrow{\pi} = \langle\{1\},\{2,3\}\rangle \rightarrow u_1(\pi_1) = 5 \geq 5;\ u_2(\pi_2) = 7 \geq 5 \Rightarrow$ MFS satisfied

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

Max-min fair share: examples

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

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

MFS evaluation:

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

---

Example: 2 agents, 1 object.

Max-min fair share: examples

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

Preferences:

	1	2	3
agent 1	5	4	2
agent 2	4	1	6

$\rightarrow u_1^{\text{MFS}} = 5$ (with cut $\langle\{1\}, \{2, 3\}\rangle$)

$\rightarrow u_2^{\text{MFS}} = 5$ (with cut $\langle\{1, 2\}, \{3\}\rangle$)

MFS evaluation:

$\overrightarrow{\pi} = \langle\{1\}, \{2, 3\}\rangle \rightarrow u_1(\pi_1) = 5 \geq 5;\ u_2(\pi_2) = 7 \geq 5 \Rightarrow$ MFS satisfied

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

---

Example: 2 agents, 1 object.

$u_1^{\text{MFS}} = u_2^{\text{MFS}} = 0 \rightarrow$ every allocation satisfies MFS!

Not very satisfactory, but can we do much better?

Max-min fair share: properties

Facts:

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

Max-min fair share: properties

Facts:

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

Max-min fair share: conjecture

Conjecture

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

Max-min fair share: conjecture

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.

Max-min fair share: conjecture

Conjecture

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

FALSE!

- 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).

Procaccia, A. D. and Wang, J. (2014).
Fair enough: Guaranteeing approximate maximin shares.
In *Proc. 14th ACM Conference on Economics and Computation (EC'14)*.
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]

Amanatidis, G., Markakis, E., Nikzad, A., and Saberi, A. (2015).
Approximation algorithms for computing maximin share allocations.
In *ICALP (1)*, volume 9134 of *Lecture Notes in Computer Science*, pages 39–51. Springer.

Kurokawa, D., Procaccia, A. D., and Wang, J. (2015).
When can the maximin share guarantee be guaranteed?
Technical report, Carnegie Mellon University.

Max-min fair share and egalitarian allocation

"Max-min fair share" sounds like "max-min optimality"...

Idea: Use the egalitarian approach to compute

$$\widehat{\vec{\pi}} = \text{argmax}_{\vec{\pi}}(min_{i \in \mathcal{A}} u_i(\pi_i))$$

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 and egalitarian allocation

"Max-min fair share" sounds like "max-min optimality"...

Idea: Use the egalitarian approach to compute

$$\widehat{\overrightarrow{\pi}} = \text{argmax}_{\overrightarrow{\pi}}(min_{i \in \mathcal{A}} u_i(\pi_i))$$

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

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.

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)

Amanatidis, G., Markakis, E., Nikzad, A., and Saberi, A. (2015).
Approximation algorithms for computing maximin share allocations.
In *ICALP (1)*, volume 9134 of *Lecture Notes in Computer Science*, pages 39–51. Springer.

Five fairness criteria

Min-max fair share

Min-max fair share

- Max-min fair share: *"I cut, you choose (I choose last)"*

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

- 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:

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

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

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"*) ?

  $\rightarrow$ Min-max fair share

Min-max fair share (mFS)

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

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

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.

Min-max fair share: properties

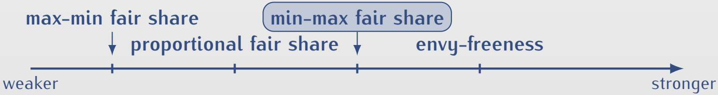
Facts:

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

Min-max fair share: properties

Facts:

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



Five fairness criteria

Competitive Equilibrium from Equal Incomes

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

Competitive Equilibrium from Equal Incomes.

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

Preferences:

	1	2	3	4
agent 1	7	2	6	10
agent 2	7	6	8	4

Competitive Equilibrium from Equal Incomes

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

Preferences:

	€0.80	€0.20	€0.80	€0.20
	1	2	3	4
agent 1	7	2	6	10
agent 2	7	6	8	4

For €1, what would you buy?

Competitive Equilibrium from Equal Incomes

Competitive Equilibrium from Equal Incomes.

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

Preferences:

	1	2	3	4
	€0.80	€0.20	€0.80	€0.20
agent 1	7	2	6	10
agent 2	7	6	8	4

For €1, what would you buy?

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

Disjoint shares!

Competitive Equilibrium from Equal Incomes

Competitive Equilibrium from Equal Incomes.

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

Preferences:

	1 (€0.80)	2 (€0.20)	3 (€0.80)	4 (€0.20)
agent 1	7	2	6	10
agent 2	7	6	8	4

For €1, what would you buy?

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

Disjoint shares!

$\Rightarrow$ Allocation $\langle\{1, 4\}, \{2, 3\}\rangle$, with prices $\langle 0.8, 0.2, 0.8, 0.2\rangle$ forms a CEEI.

$\Rightarrow$ Allocation $\langle\{1, 4\}, \{2, 3\}\rangle$ satisfies CEEI.

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.

Moulin, H. (1995).
Cooperative Microeconomics, A Game-Theoretic Introduction.
Prentice Hall.

Othman, A., Sandholm, T., and Budish, E. (2010).
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.

CEEI: known facts

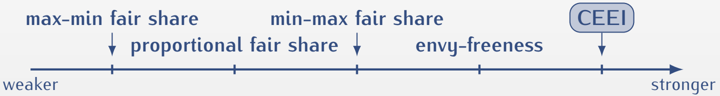
Fact: $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.

CEEI: known facts

Fact: $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.

CEEI: known facts

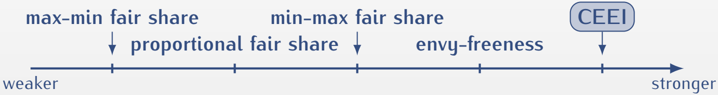
Fact: $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.



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

CEEI: known facts

Fact: $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.



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

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

CEEI: known facts

Fact: $\vec{\pi}$ satisfies CEEI $\Rightarrow$ $\vec{\pi}$ is envy-free.

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 $(\vec{\pi}, \vec{p})$ a CEEI? $\rightarrow$ **coNP**-complete
- Does there exist a CEEI? **NP**-hard

Brânzei, S., Hosseini, H., and Miltersen, P. B. (2015).
Characterization and computation of equilibria for indivisible goods.
In *Algorithmic Game Theory*, pages 244–255. Springer.

CEEI: open problems

- Does there exist a CEEI? **NP**-hard and in Σ_2^P. Precise complexity?
- How to test whether $\vec{\pi}$ is a CEEI (and find the associated $\vec{p}$)?

CEEI: open problems

- Does there exist a CEEI? **NP**-hard and in Σ_2^P. Precise complexity?
- How to test whether $\overrightarrow{\pi}$ is a CEEI (and find the associated $\overrightarrow{p}$)?

$$0 \leq \mathbf{p_o} \leq 1, \text{ for all } o \in [\![1, m]\!] \tag{1}$$

$$\sum_{o=1}^{m} a_{\pi_i,o}\mathbf{p_o} \leq 1, \text{ for all } i \in [\![1, n]\!], \text{with } a_{\pi_i,o} = 1 \text{ if } o \in \pi_i, \text{ 0 otherwise} \tag{2}$$

$$\sum_{o=1}^{m} a_{\pi',o}\mathbf{p_o} > 1, \text{ for all } \pi' \text{ such that } \exists i \text{ such that } u_i(\pi') > u_i(\pi_i) \tag{3}$$

CEEI: open problems

- Does there exist a CEEI? **NP**-hard and in Σ_2^P. Precise complexity?
- How to test whether $\overrightarrow{\pi}$ is a CEEI (and find the associated $\overrightarrow{p}$)?

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

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

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

CEEI: open problems

- Does there exist a CEEI? **NP**-hard and in Σ_2^P. Precise complexity?
- How to test whether $\overrightarrow{\pi}$ is a CEEI (and find the associated $\overrightarrow{p}$)?

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

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

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

- How to compute a CEEI allocation?

CEEI: open problems

- Does there exist a CEEI? **NP**-hard and in Σ_2^P. Precise complexity?
- How to test whether $\overrightarrow{\pi}$ is a CEEI (and find the associated $\overrightarrow{p}$)?

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

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

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

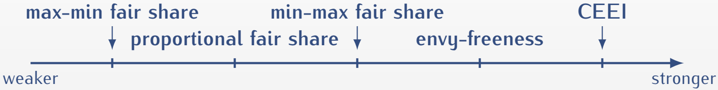
- How to compute a CEEI allocation?

Simplistic algorithm: compute all allocations and test which ones are CEEI.

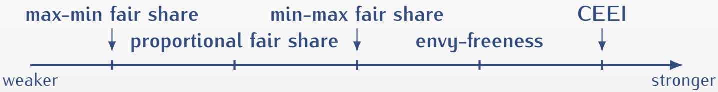
Five fairness criteria

Summary and interpretation

Interpretation



Interpretation

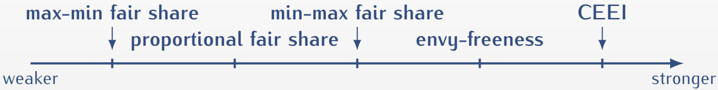


1 For all allocation $\overrightarrow{\pi}$:

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

$\rightarrow$ the highest property $\overrightarrow{\pi}$ satisfies, the most satisfactory it is.

Interpretation



1 For all allocation $\vec{\pi}$:

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

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

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

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

$\rightarrow$ the lowest subset, the less “conflict-prone”.

Interpretation

weaker → stronger: max-min fair share, proportional fair share, min-max fair share, envy-freeness, CEEI

1 For all allocation $\overrightarrow{\pi}$:

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

$\rightarrow$ the highest property $\overrightarrow{\pi}$ satisfies, the most satisfactory it is.

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

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

$\rightarrow$ the lowest subset, the less “conflict-prone”.

Two extreme examples:

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

	1	2
agent 1	1000	0
agent 2	0	1000

$\rightarrow$ in $\mathcal{I}_{|\text{CEEI}}$ (with *e.g.* $\overrightarrow{p} = \langle 1, 1 \rangle$).

A glimpse at experiments

What about fairness criteria in practice?

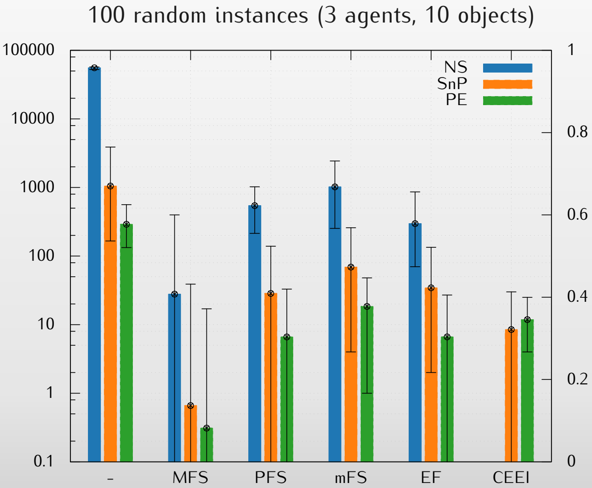
A glimpse at experiments

What about fairness criteria in practice?

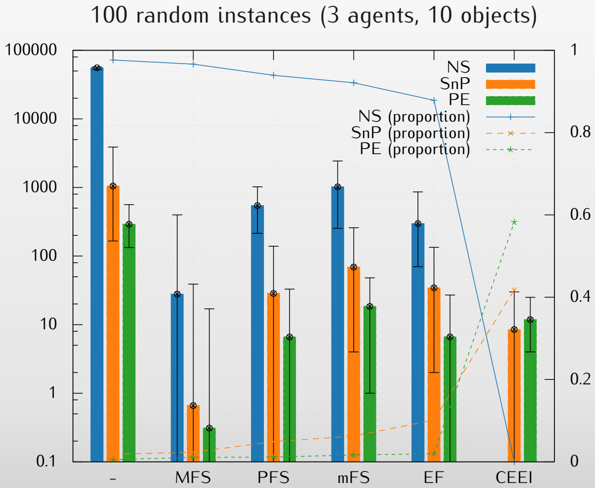
Goal of our experiments: evaluate the distribution of the allocation over:

- the fairness scale (–, MFS, PFS, mFS, EF, CEEI);
- three efficiency levels (–, Sequenceable, Pareto-efficient).

A glimpse at experiments



A glimpse at experiments



k-additive preferences

- Additive preferences are nice but have a limited expressiveness.

k-additive preferences

- 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)

k-additive preferences

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

k-additive preferences

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

k-additive preferences

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

k-additive preferences

A weight $w(\mathcal{S})$ to each subset $\mathcal{S}$ 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)
    $\rightarrow u(\{skis, poles\}) > u(skis) + u(poles)$
  - the pair of skis and the snowboard (substitutability)
    $\rightarrow u(\{skis, snowboard\}) < u(skis) + u(snowboard)$

k-additive preferences

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

Examples:

- $w(skis) = 10$; $w(poles) = 0$; $w(\{skis, poles\}) = 90$
  $\rightarrow u(\{skis, poles\}) = 100 > 10 + 0$
- $w(skis) = 100$; $w(snowboard) = 100$; $w(\{skis, snowboard\}) = -100$
  $\rightarrow u(\{skis, snowboard\}) = 100 < 100 + 100$

MFS and k-additive preferences

Reminder

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

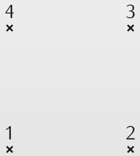
MFS and k-additive preferences

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



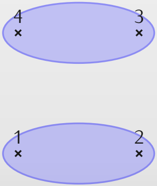
MFS and k-additive preferences

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

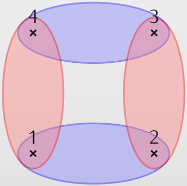
MFS and k-additive preferences

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

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

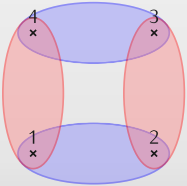
MFS and k-additive preferences

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

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

Worse... Deciding whether there exists one is **NP**-complete [PARTITION].

Take-away message

A scale of properties (for numerical additive preferences)...

Take-away message

A scale of properties (for numerical additive preferences)...

Max-min fair share
Almost always possible to satisfy it

Take-away message

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

Take-away message

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

Take-away message

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

Take-away message

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

Take-away message

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.

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...)