Positional Scoring Rules for the Allocation of Indivisible Goods

Dorothea Baumeister, Trung Thanh Nguyen, Jörg Rothe
Heinrich-Heine Universität Düsseldorg

Sylvain Bouveret
LIG – Grenoble INP

Jérôme Lang, Abdallah Saffidine
LAMSADE, Université Paris Dauphine

Eleventh European Workshop on Multi-agent Systems

Toulouse, December 12–13, 2013

Fair division of indivisible goods

Fair division of indivisible goods…

We 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

Fair division of indivisible goods

Fair division of indivisible goods...

We have:

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

We 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

Fair division of indivisible goods

Fair division of indivisible goods…

We 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

We 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

Plenty of real-world applications: course allocation, operation of Earth observing satellites, …

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 maximizes the **social welfare**,

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 maximizes the **social welfare**,
  - $\sum_{i \in \mathcal{A}} u_i(\pi)$ – classical utilitarian solution

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 maximizes the **social welfare**,
  - $\sum_{i \in \mathcal{A}} u_i(\pi)$ – classical utilitarian solution
  - $\min_{i \in \mathcal{A}} u_i(\pi)$ – egalitarian solution [Bansal and Sviridenko, 2006]

Bansal, N. and Sviridenko, M. (2006).
The Santa Claus problem.
In *Proceedings of STOC'06*, pages 31–40. ACM Press.

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

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

- Find an allocation $\overrightarrow{\pi}$ that maximizes the **social welfare**,
  - $\sum_{i \in \mathcal{A}} u_i(\pi)$ – classical utilitarian solution
  - $\min_{i \in \mathcal{A}} u_i(\pi)$ – egalitarian solution [Bansal and Sviridenko, 2006]
  - the lexicographic minimum over $(u_1(\pi), \dots, u_n(\pi))$ – refinement of the egalitarian solution

Bansal, N. and Sviridenko, M. (2006).
The Santa Claus problem.
In *Proceedings of STOC'06*, pages 31–40. ACM Press.

Example

Example: 3 objects $\{o_1, o_2, o_3\}$, 2 agents $\{1, 2\}$.

Example

Example: 3 objects $\{o_1, o_2, o_3\}$, 2 agents $\{1, 2\}$.

Preferences:

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Example

Example: 3 objects $\{o_1, o_2, o_3\}$, 2 agents $\{1, 2\}$.

Preferences:

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Classical utilitarian evaluation:

$$\overrightarrow{\pi} = \langle \{o_1\}, \{o_2, o_3\} \rangle \rightarrow uc(\overrightarrow{\pi}) = 5 + (4 + 6) = 15$$

Example

Example: 3 objects $\{o_1, o_2, o_3\}$, 2 agents $\{1, 2\}$.

Preferences:

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Classical utilitarian evaluation:

$\overrightarrow{\pi} = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}) = 5 + (4 + 6) = 15$

$\overrightarrow{\pi}' = \langle\{o_1, o_2\}, \{o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}') = (5 + 1) + 6 = 11$

Example

Example: 3 objects $\{o_1, o_2, o_3\}$, 2 agents $\{1, 2\}$.

Preferences:

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Classical utilitarian evaluation:

$\overrightarrow{\pi} = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}) = 5 + (4 + 6) = 15$

$\overrightarrow{\pi}' = \langle\{o_1, o_2\}, \{o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}') = (5 + 1) + 6 = 11$

Egalitarian evaluation:

$\overrightarrow{\pi} = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}) = \min(5, 6 + 4) = 5$

Example

Example: 3 objects $\{o_1, o_2, o_3\}$, 2 agents $\{1, 2\}$.

Preferences:

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Classical utilitarian evaluation:

$\overrightarrow{\pi} = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}) = 5 + (4 + 6) = 15$

$\overrightarrow{\pi}' = \langle\{o_1, o_2\}, \{o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}') = (5 + 1) + 6 = 11$

Egalitarian evaluation:

$\overrightarrow{\pi} = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}) = \min(5, 6 + 4) = 5$

$\overrightarrow{\pi}' = \langle\{o_1, o_2\}, \{o_3\}\rangle \rightarrow uc(\overrightarrow{\pi}') = \min(5 + 1, 6) = 6$

About ordinal preferences

Problem: This approach requires **numerical** preferences.

About ordinal preferences

Problem: This approach requires **numerical** preferences.

$\rightsquigarrow$ Meaningless if no clear numerical scale (*e.g* money) is involved...

About ordinal preferences

Problem: This approach requires **numerical** preferences.
$\leadsto$ Meaningless if no clear numerical scale (*e.g* money) is involved...

Our starting point: *What can we do with ordinal preferences?*

About scoring vectors

Here we take inspiration from **voting theory**.

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring functions:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring functions:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1
Lexicographic	32	16	8	4	2	1

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring functions:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1
Lexicographic	32	16	8	4	2	1
Quasi-Indifference	$1 + 5\varepsilon$	$1 + 4\varepsilon$	$1 + 3\varepsilon$	$1 + 2\varepsilon$	$1 + \varepsilon$	1

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring functions:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1
Lexicographic	32	16	8	4	2	1
Quasi-Indifference	$1 + 5\varepsilon$	$1 + 4\varepsilon$	$1 + 3\varepsilon$	$1 + 2\varepsilon$	$1 + \varepsilon$	1
k-Approval	1	1	0	0	0	0

Example

Example

5 objects, 3 agents...

- $1 : o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- $2 : o_4 \succ o_2 \succ o_5 \succ o_1 \succ o_3$
- $3 : o_1 \succ o_3 \succ o_5 \succ o_4 \succ o_2$

Example

Example

5 objects, 3 agents...

- $1 : o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- $2 : o_4 \succ o_2 \succ o_5 \succ o_1 \succ o_3$
- $3 : o_1 \succ o_3 \succ o_5 \succ o_4 \succ o_2$

Let's consider allocation $\pi = \langle \{o_1\}, \{o_4, o_2\}, \{o_3, o_5\} \rangle$.

Example

Example

5 objects, 3 agents...

- $1 : o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- $2 : o_4 \succ o_2 \succ o_5 \succ o_1 \succ o_3$
- $3 : o_1 \succ o_3 \succ o_5 \succ o_4 \succ o_2$

Let's consider allocation $\pi = \langle \{o_1\}, \{o_4, o_2\}, \{o_3, o_5\} \rangle$.

- **Borda:** $u_1(\pi) = 5$; $u_2(\pi) = 5 + 4 = 9$; $u_3(\pi) = 4 + 3 = 7$.
- **Lexicographic:** $u_1(\pi) = 16$; $u_2(\pi) = 24$; $u_3(\pi) = 12$.
- **QI:** $u_1(\pi) = 1 + 4\varepsilon$; $u_2(\pi) = 2 + 7\varepsilon$; $u_3(\pi) = 2 + 5\varepsilon$.
- **2-approval:** $u_1(\pi) = 1$; $u_2(\pi) = 2$; $u_3(\pi) = 1$.

Positional scoring allocation rules

Back to our resource allocation problem...

Assume that the agents have **ordinal** preferences (rankings).

Positional scoring allocation rules

Back to our resource allocation problem...

Assume that the agents have **ordinal** preferences (rankings).

Interpretation:
Borda SF
Lexicographic SF
Quasi-Indifference SF
k-Approval SF

Positional scoring allocation rules

Back to our resource allocation problem...

Assume that the agents have **ordinal** preferences (rankings).

Maximize:	$\sum_i u_i(\pi)$
Interpretation:	
Borda SF	
Lexicographic SF	
Quasi-Indifference SF	
k-Approval SF	

Positional scoring allocation rules

Back to our resource allocation problem...
Assume that the agents have **ordinal** preferences (rankings).

Maximize: Interpretation:	$\sum_i u_i(\pi)$	$\min_i u_i(\pi)$
Borda SF		
Lexicographic SF		
Quasi-Indifference SF		
k-Approval SF		

Positional scoring allocation rules

Back to our resource allocation problem...

Assume that the agents have **ordinal** preferences (rankings).

Maximize: Interpretation:	$\sum_i u_i(\pi)$	$\min_i u_i(\pi)$	$\text{leximin}(u_1(\pi), \ldots, u_n(\pi))$
Borda SF			
Lexicographic SF			
Quasi-Indifference SF			
k-Approval SF			

Positional scoring allocation rules

Back to our resource allocation problem...
Assume that the agents have **ordinal** preferences (rankings).

Maximize: / Interpretation:	$\sum_i u_i(\pi)$	$\min_i u_i(\pi)$	$\text{leximin}(u_1(\pi), \ldots, u_n(\pi))$
Borda SF			
Lexicographic SF			
Quasi-Indifference SF			
k-Approval SF			

$\rightsquigarrow$ 12 **positional scoring allocation rules**
(transposition to resource allocation of positional scoring rules in voting)

The problems studied

We have 12 allocation rules, having nice properties (*e.g* monotonicity), but what is their precise complexity?

The problems studied

We have 12 allocation rules, having nice properties (*e.g* monotonicity), but what is their precise complexity?

For each pair (scoring vector, social criterion), what is the complexity of...

1. **Optimal Allocation Value (OAV):** is it possible to find an allocation of utility $\geq K$?
2. **Optimal Allocation (OA):** does π belong to the set of optimal allocations?
3. **Find Optimal Allocation (FOA):** find an optimal allocation.

Main results

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!

Idea: give each item to the agent that ranks it the best.

Main results

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!

Idea: give each item to the agent that ranks it the best.

For $\min_i u_i(\pi)$ (egalitarianism):

- **Bad news:** **hard** (NP-complete, coNP-complete, NP-hard for OAV, OA, FOA resp.) for Borda, lexicographic and QI scoring functions.

(all by reduction from [X3C])

Main results

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!

Idea: give each item to the agent that ranks it the best.

For $\min_i u_i(\pi)$ (egalitarianism):

- **Bad news:** hard (NP-complete, coNP-complete, NP-hard for OAV, OA, FOA resp.) for Borda, lexicographic and QI scoring functions.

  (all by reduction from [X3C])

- **Good news:** easy (polynomial)
  - if the number of objects is fixed (obvious);
  - if the number of agents is fixed (dynamic programming);
  - for k-approval ("known" problem).

Main results

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!

Idea: give each item to the agent that ranks it the best.

For $\min_i u_i(\pi)$ (egalitarianism):

- **Bad news:** hard (NP-complete, coNP-complete, NP-hard for OAV, OA, FOA resp.) for Borda, lexicographic and QI scoring functions.

  (all by reduction from [X3C])

- **Good news:** easy (polynomial)
  - if the number of objects is fixed (obvious);
  - if the number of agents is fixed (dynamic programming);
  - for k-approval ("known" problem).

Most results for min carry over to leximin.

Results: summary

	OA	OAV	FOA
$F_{s,+}$	in P	in P	pol. time
$F_{s,\min}$	coNP-comp*	NP-comp*	NP-hard*
k-app or $m \in O(1)$	in P	in P	pol. time
lex or ε-qi	coNP-comp	NP-comp	NP-hard
borda	coNP-comp	NP-comp	
lex or borda or ε-qi, if $n \in O(1)$	in P	in P	pol. time
$F_{s,\text{leximin}}$	coNP-comp*	NP-comp*	NP-hard*
lex or ε-qi	in coNP	NP-comp	NP-hard
borda	in coNP	NP-comp	
lex or borda or ε-qi, if $n \in O(1)$	in P	in P	pol. time

About approximation

Most cases are **hard**...

Question: *Is it possible to efficiently compute* ***good*** *(but potentially suboptimal) allocations?*

About approximation

Most cases are **hard**...

Question: *Is it possible to efficiently compute **good** (but potentially suboptimal) allocations?*

Our approach: Instead of giving general approximation results[1], we:

- focus on a **simple** allocation protocol;
- and try to analyze how good the allocations it gives are.

[1] Actually there is one in the paper, for (lexico, min).

An elicitation-free protocol...

Ask the agents to pick in turn their most preferred object among the remaining ones, according to some **predefined sequence** σ.

Example

3 agents 1, 2, 3 / 6 objects / sequence 123321 $\rightarrow$ 1 chooses first (and takes her preferred object), then 2, then 3, then 3 again...

An elicitation-free protocol...

Ask the agents to pick in turn their most preferred object among the remaining ones, according to some **predefined sequence** σ.

Example

3 agents 1, 2, 3 / 6 objects / sequence 123321 $\rightarrow$ 1 chooses first (and takes her preferred object), then 2, then 3, then 3 again...

Arguably a very simple (and natural) protocol! **No elicitation required**!

An elicitation-free protocol...

Ask the agents to pick in turn their most preferred object among the remaining ones, according to some **predefined sequence** σ.

Example

3 agents 1, 2, 3 / 6 objects / sequence 123321 $\rightarrow$ 1 chooses first (and takes her preferred object), then 2, then 3, then 3 again...

Arguably a very simple (and natural) protocol! **No elicitation required**!

Here we focus on **regular sequences** σ of the kind $(1 \ldots n)^*$, but our results are similar for alternating sequences like $(1 \ldots nn \ldots 1)^*$.

Price of elicitation-freeness

Arguably a very simple (and natural) protocol...

Price of elicitation-freeness

Arguably a very simple (and natural) protocol...

...but obviously suboptimal!

Price of elicitation-freeness

Arguably a very simple (and natural) protocol…

…but obviously suboptimal!

Question: *What is the loss in social welfare we incur by using such a simple protocol instead of computing the optimal allocation?*

Price of elicitation-freeness

Arguably a very simple (and natural) protocol...

...but obviously suboptimal!

Question: *What is the loss in social welfare we incur by using such a simple protocol instead of computing the optimal allocation?*

More formally:

- **Multiplicative** Price of Elicitation-Freeness: worst case ratio $u_c^{\text{opt}}/u_c(\sigma)$, for a sequence σ.
- **Additive** Price of Elicitation-Freeness: worst case difference $u_c^{\text{opt}} - u_c(\sigma)$, for a sequence σ.

Some theoretical bounds for Borda

For **classical utilitarianism** ($\sum_i u_i(\pi)$):

$$1 + \frac{n-1}{m} + \Theta(\frac{1}{m^2}) \leq MPEF \leq 2 - \frac{1}{n} + \Theta(\frac{1}{m^2}), \text{ when } m \to +\infty.$$

Some theoretical bounds for Borda

For **classical utilitarianism** ($\sum_i u_i(\pi)$):

$$1 + \frac{n-1}{m} + \Theta(\frac{1}{m^2}) \leq MPEF \leq 2 - \frac{1}{n} + \Theta(\frac{1}{m^2}), \text{ when } m \to +\infty.$$

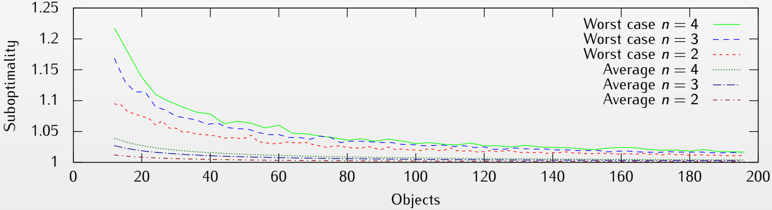
For **egalitarianism** ($\min_i u_i(\pi)$):

$$MPEF \leq 2 - \frac{1}{n} + \Theta(\frac{1}{m^2}), \text{ when } m \to +\infty.$$

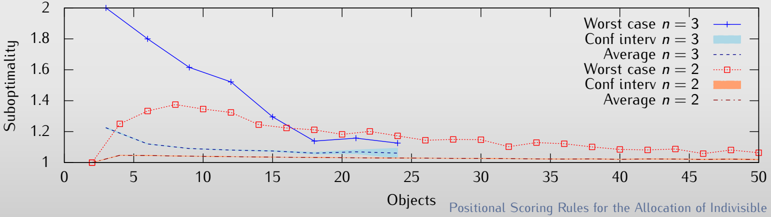
See the paper for more!

Some experimental results for Borda

For **classical utilitarianism** ($\sum_i u_i(\pi)$):



For **egalitarianism** ($\min_i u_i(\pi)$):



Take-away message

This work is about **allocating indivisible objects to agents**.

Take-away message

This work is about **allocating indivisible objects to agents**.

Our starting point: most allocation procedures rely on a **numerical elicitation** of preferences which is uneasy for the agents. $\rightarrow$ *What can we do with* ***ordinal preferences****?*

Take-away message

This work is about **allocating indivisible objects to agents**.

Our starting point: most allocation procedures rely on a **numerical elicitation** of preferences which is uneasy for the agents. $\rightarrow$ *What can we do with **ordinal preferences**?*

Positional scoring allocation rules

Take-away message

This work is about **allocating indivisible objects to agents**.

Our starting point: most allocation procedures rely on a **numerical elicitation** of preferences which is uneasy for the agents. $\rightarrow$ *What can we do with **ordinal preferences**?*

Positional scoring allocation rules

- Characterization of the computational complexity (mostly **hard**)
- Approximation with a very simple sequential protocol.

Take-away message

This work is about **allocating indivisible objects to agents**.

Our starting point: most allocation procedures rely on a **numerical elicitation** of preferences which is uneasy for the agents. $\rightarrow$ *What can we do with **ordinal preferences**?*

Positional scoring allocation rules

- Characterization of the computational complexity (mostly **hard**)
- Approximation with a very simple sequential protocol.

Possible future work: manipulation, link with envy-freeness...