Winter School and Workshop on Algorithmic Game Theory
Singapore, 14-18 Jan 2013

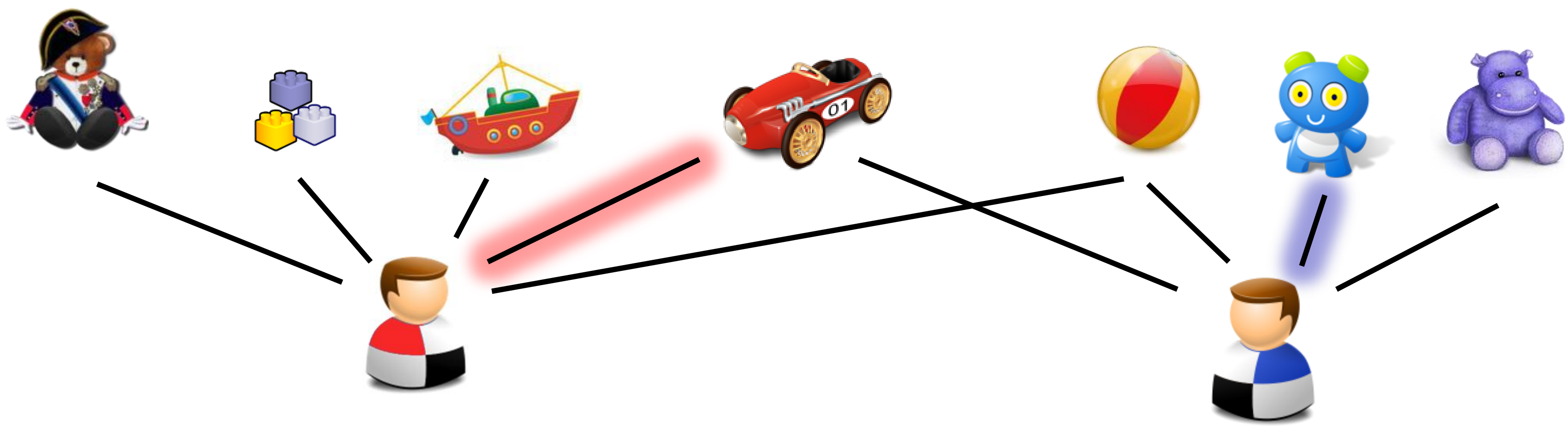
Mechanism for Fair Allocations of Indivisible Goods

No-Punishment Payment Rules in Fully Verifiable Settings

joint work with Francesco Scarcello

Gianluigi Greco
University of Calabria

The Model

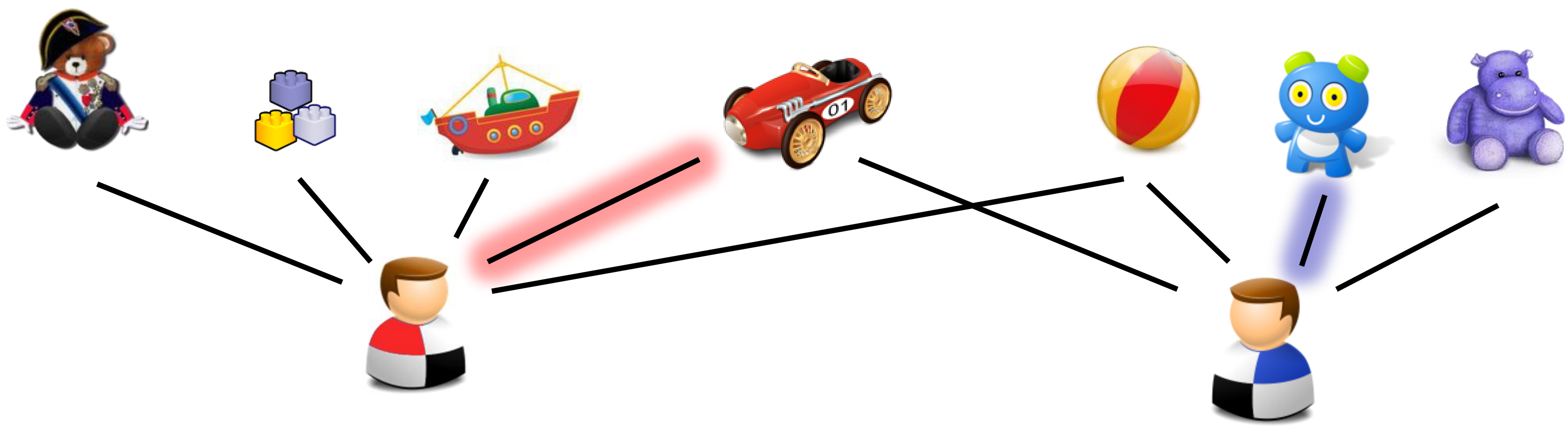


- Goods are indivisible and non-sharable
- Constraints on the max/min number of goods to be allocated to each agent
- Agent preferences: *Private* types VS *Declared* types

  - Monetary compensation to induce truthfulness

see, e.g., [Shoham, Leyton-Brown; 2009]

The Model



- Goods are indivisible and non-sharable
- Constraints on the max/min number of goods to be allocated to each agent
- Agent preferences: *Private* types VS *Declared* types

- Monetary compensation to induce truthfulness

  «budget balance»

  - The algebraic sum of the monetary transfers is zero
  - In particular, mechanisms cannot run into deficit

Goals of the Allocation

- «Efficiency»
  - Maximize the social welfare
- «Fairness»
  - For instance, it is desirable that ***no agent envies*** the allocation of any another agent, or that
  - the selected outcome is ***Pareto efficient***, i.e., there must be no different allocation such that every agent gets at least the same utility and one of them even improves.

see, e.g., [Brandt, Endriss; 2012]

(A Few…) Impossibility Results

- Efficiency **+** Truthfulness **+** Budget Balance

  [Green, Laffont; 1977]
  [Hurwicz; 1975]

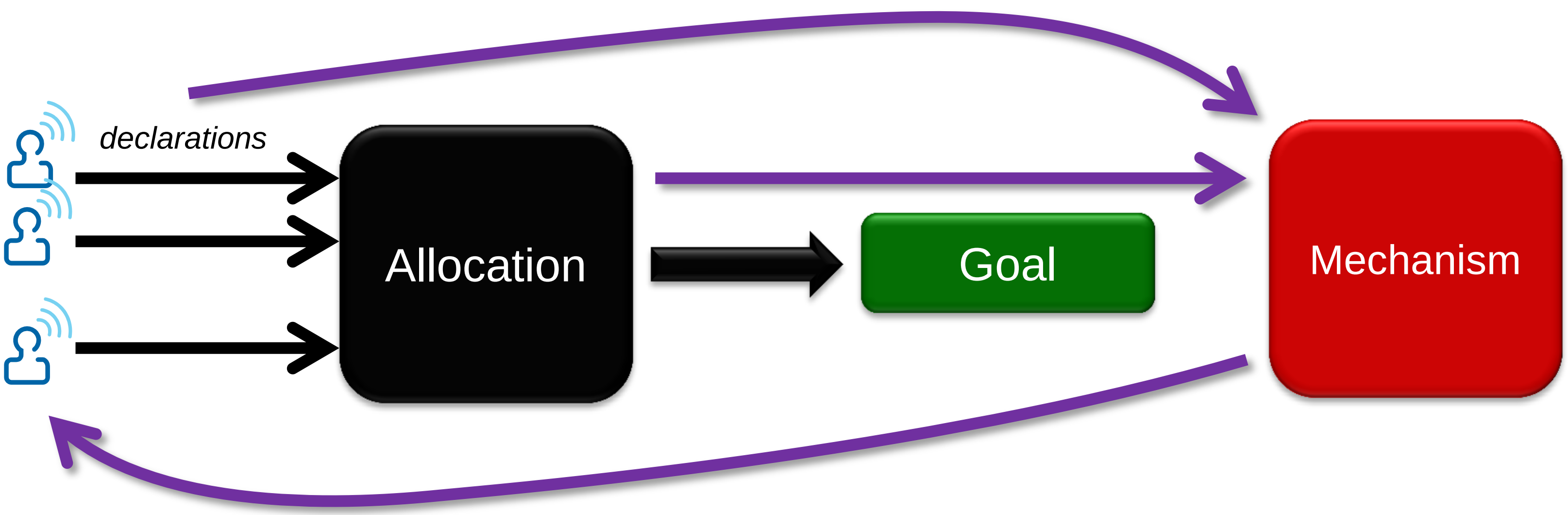
- Fairness **+** Truthfulness **+** Budget Balance

  [Tadenuma, Thomson;1995]
  [Alcalde, Barberà; 1994]
  [Andersson, Svensson, Ehlers; 2010]

(A Few…) Impossibility Results

Efficiency **+** Truthfulness **+** Budget Balance

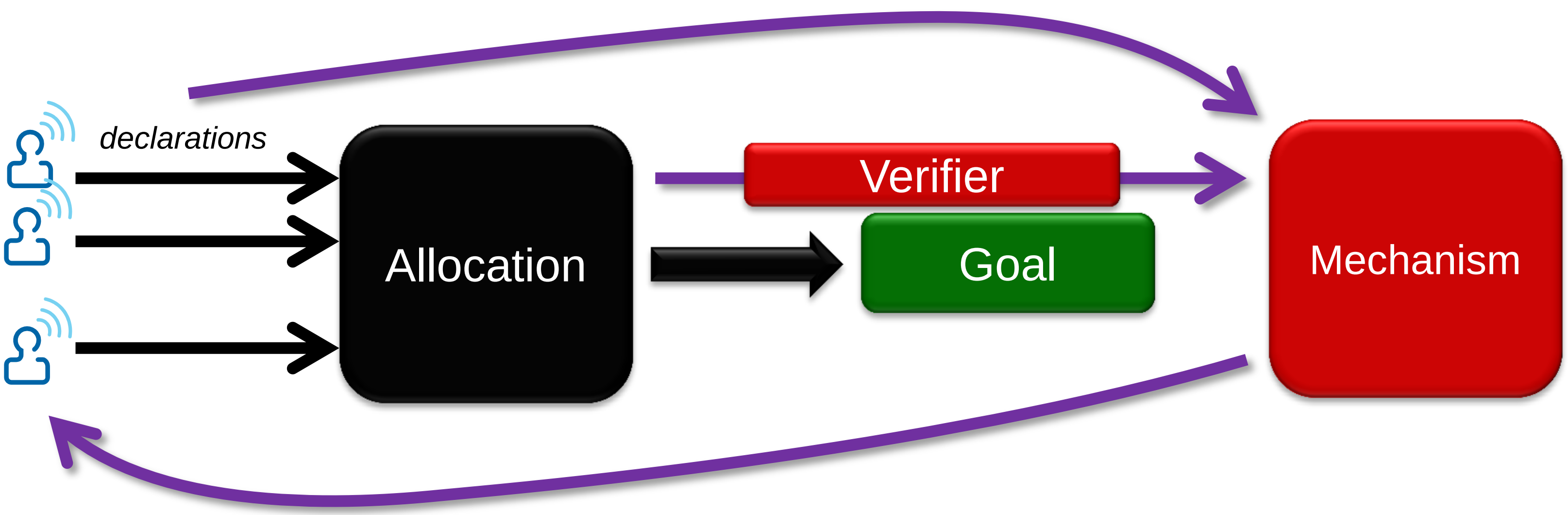
Fairness **+** Truthfulness **+** Budget Balance



(A Few…) Impossibility Results

- Efficiency **+** Truthfulness **+** Budget Balance
- Fairness **+** Truthfulness **+** Budget Balance

- Verification on «selected» declarations



Approaches to Verification

(1) Partial Verification

[Green, Laffont; 1986]
[Nisan, Ronen; 2001]

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

Approaches to Verification

(1) Partial Verification

[Auletta, De Prisco, Ferrante, Krysta, Parlato, Penna, Persiano, Sorrentino, Ventre]

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

Approaches to Verification

(1) Partial Verification

[Auletta, De Prisco, Ferrante, Krysta, Parlato, Penna, Persiano, Sorrentino, Ventre]

(2) Probabilistic Verification

[Caragiannis, Elkind, Szegedy, Yu; 2012]

Punishments are used to enforce truthfulness

Approaches to Verification

(1) Partial Verification

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

- Partial verification to guarantee fairness [not covered here]

Approaches to Verification

(1) Partial Verification

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

- Partial verification to guarantee fairness [not covered here]

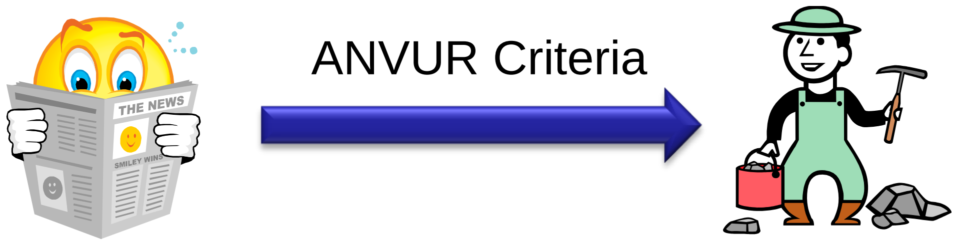
(3) Full Verification

Outline

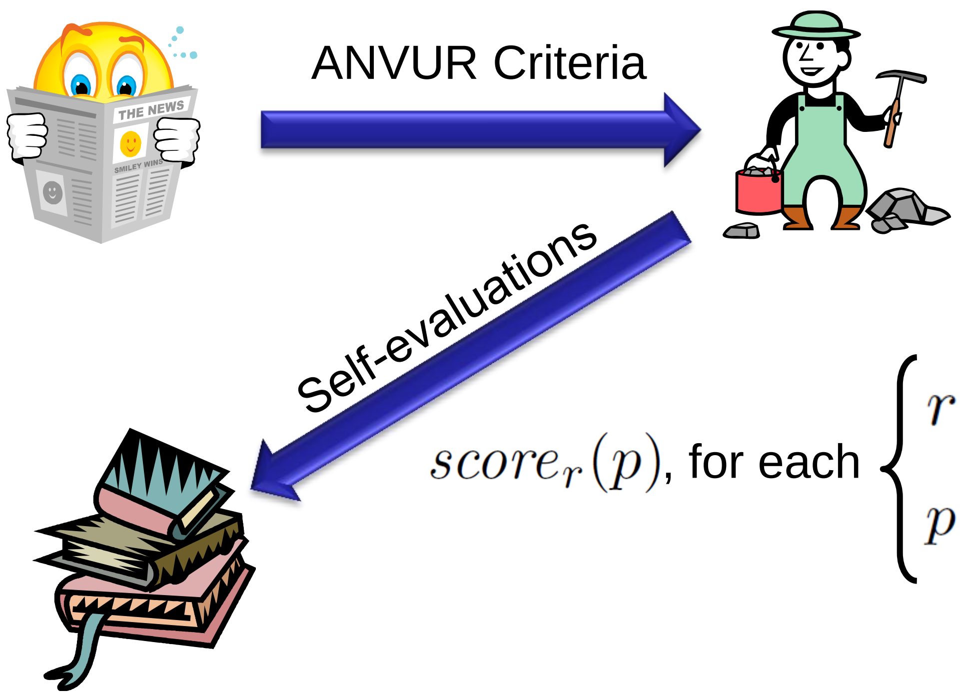
Case study: Italian Research Assessment Program

- VQR 2004-2010: ANVUR should evaluate the quality of research of all Italian research structures
- Funds for the structures in the next years depend on the outcome of this evaluation
- Substructures will be also evaluated (e.g. university departments)

ANVUR Evaluation

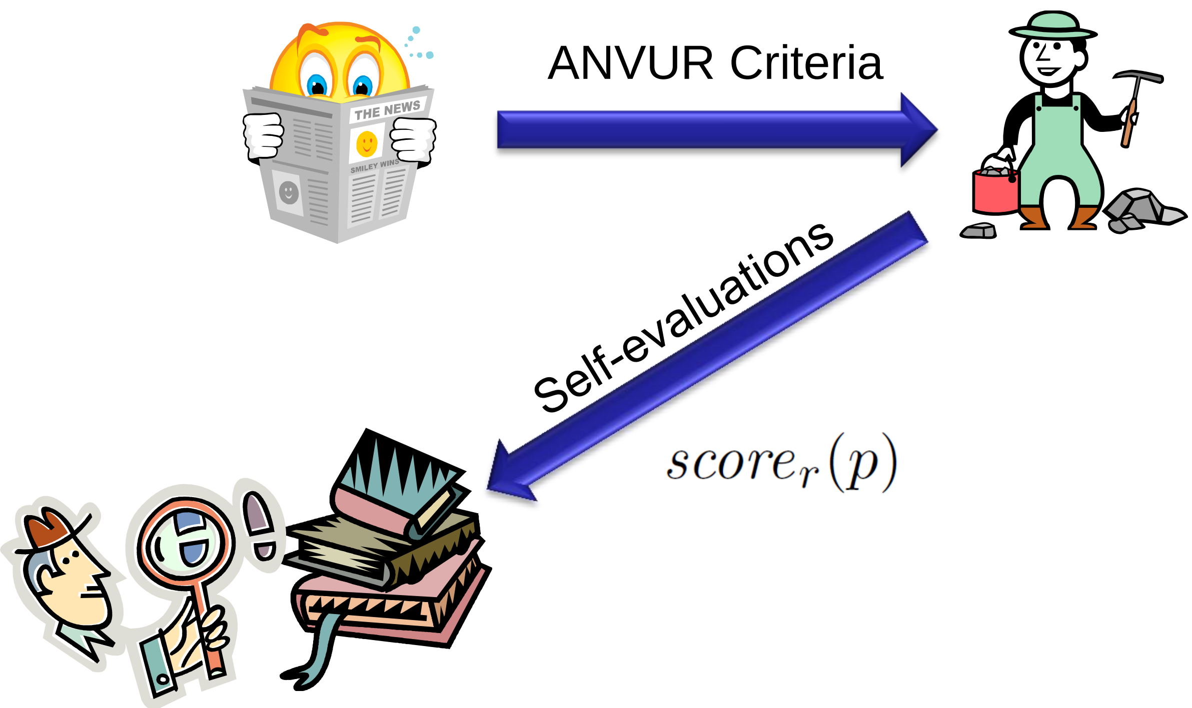


ANVUR Evaluation



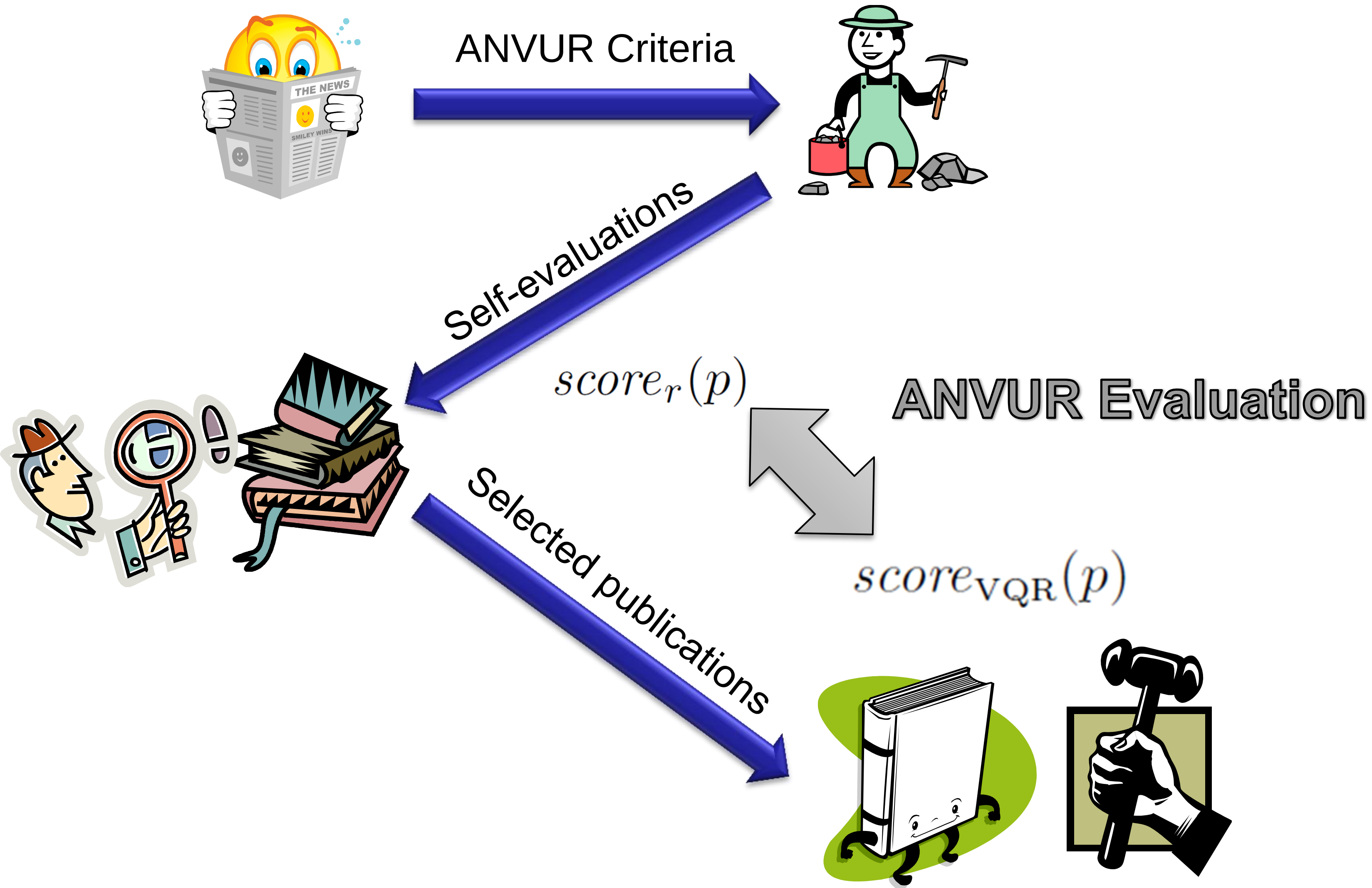
$$score_r(p), \text{ for each } \begin{cases} r \in \mathcal{R} \\ p \in products(r) \end{cases}$$

ANVUR Evaluation

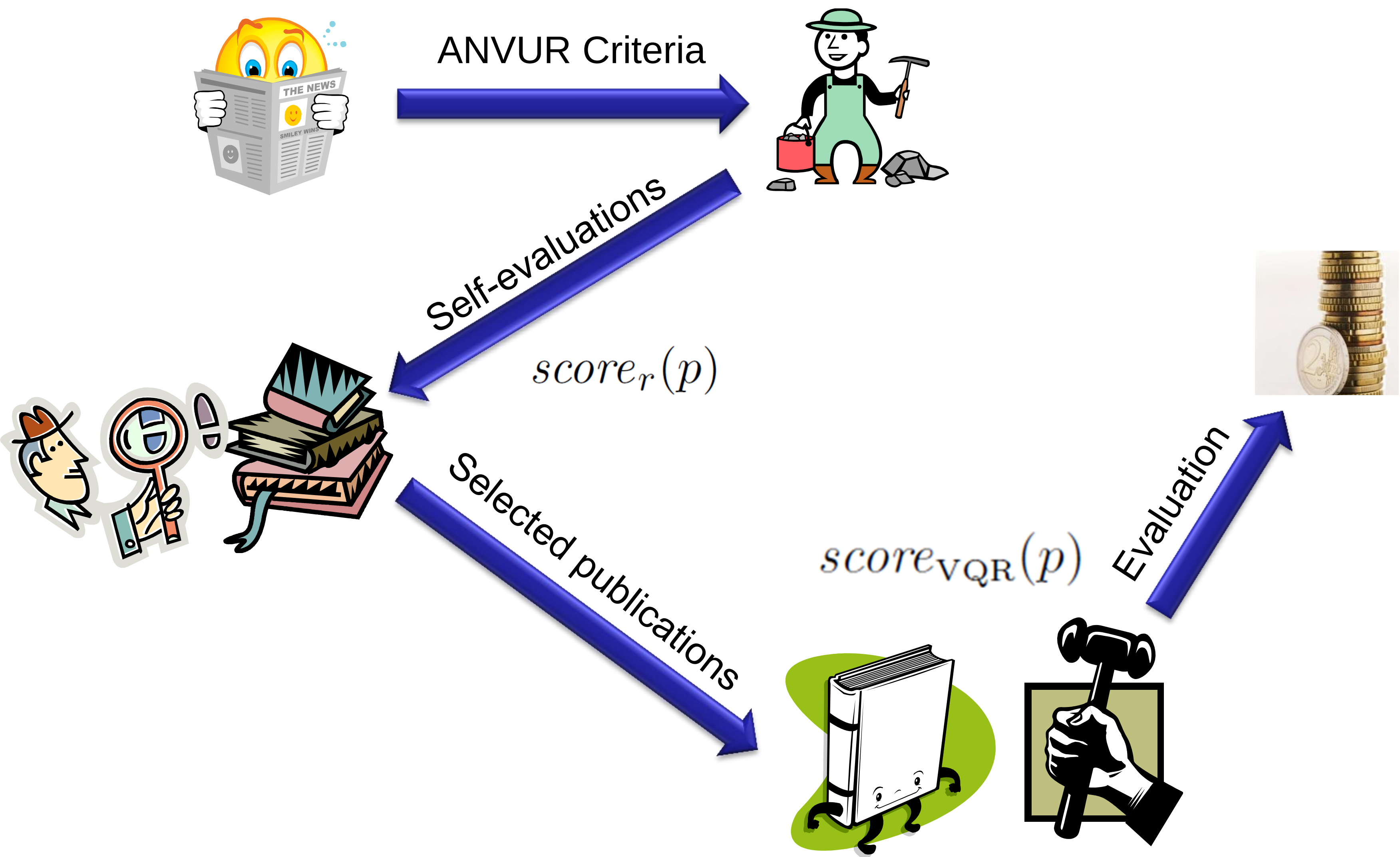


Structures are in charge of selecting the products to submit

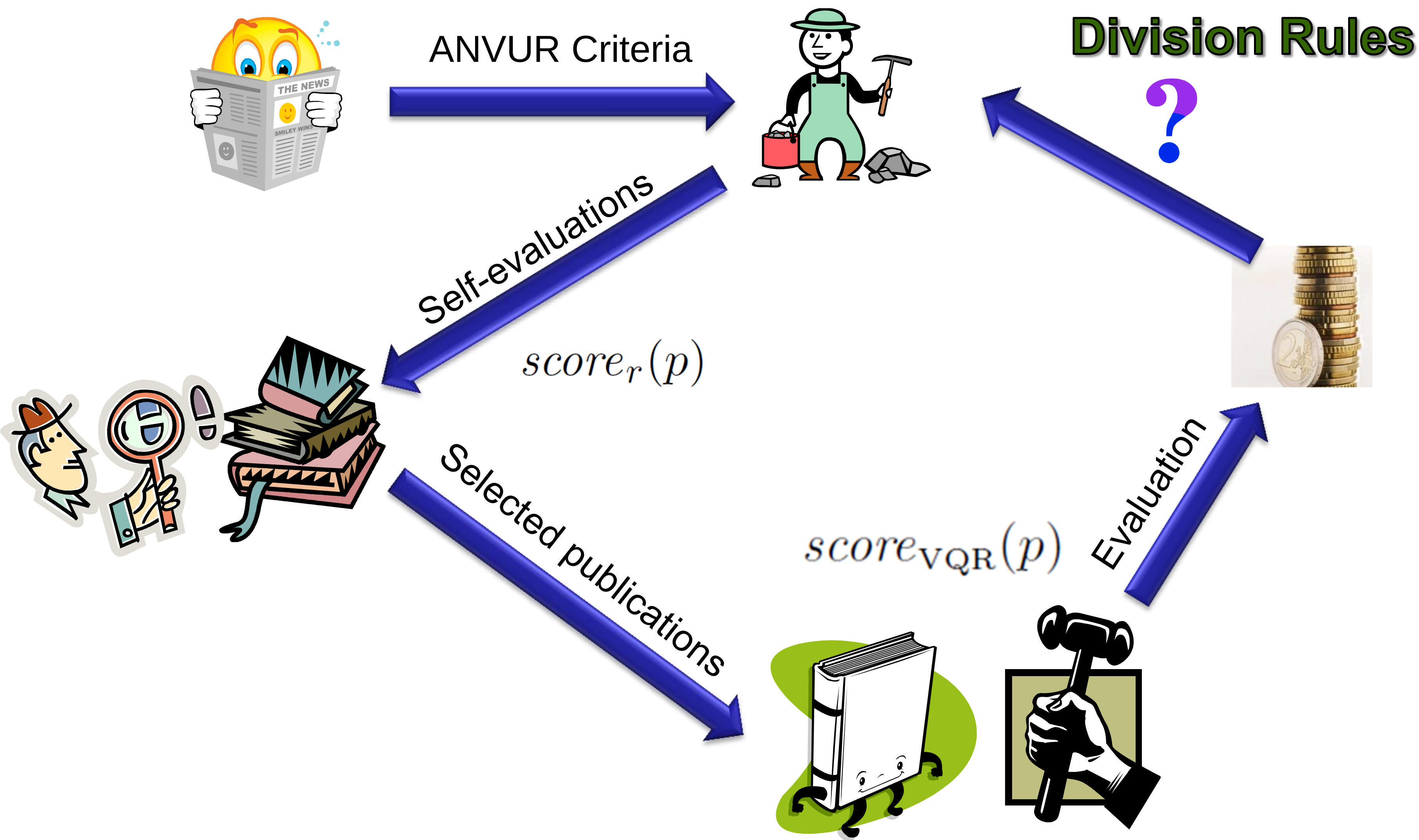
ANVUR Evaluation



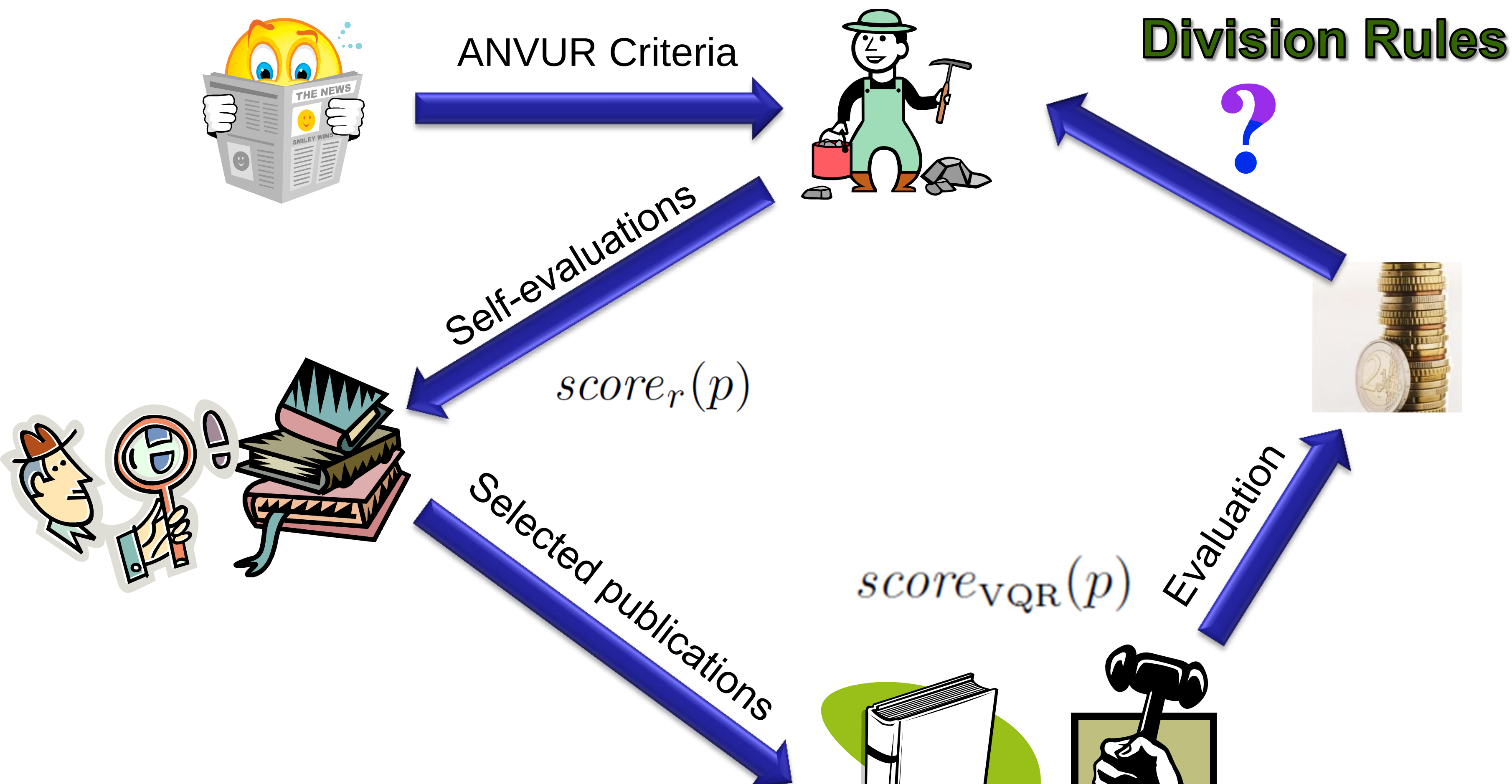
ANVUR Evaluation



ANVUR Evaluation

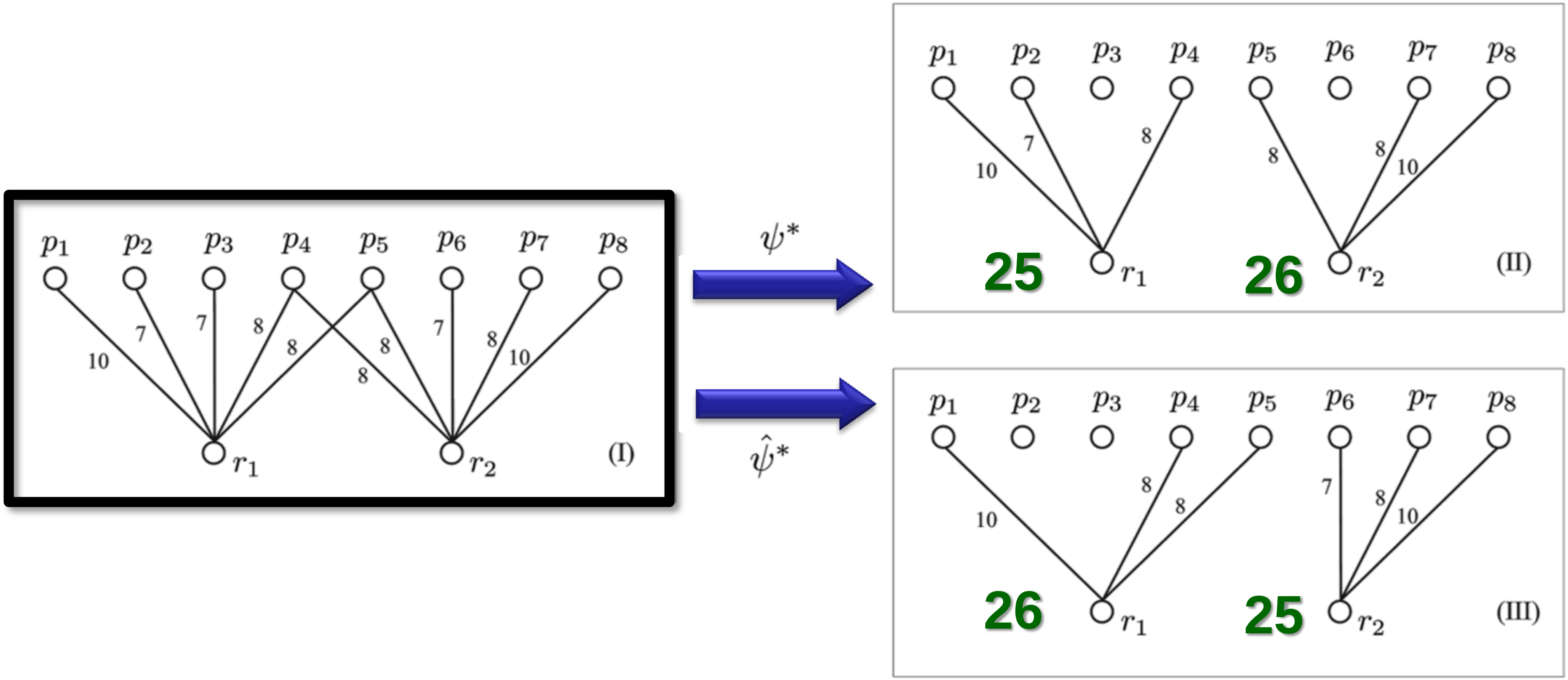


ANVUR Evaluation

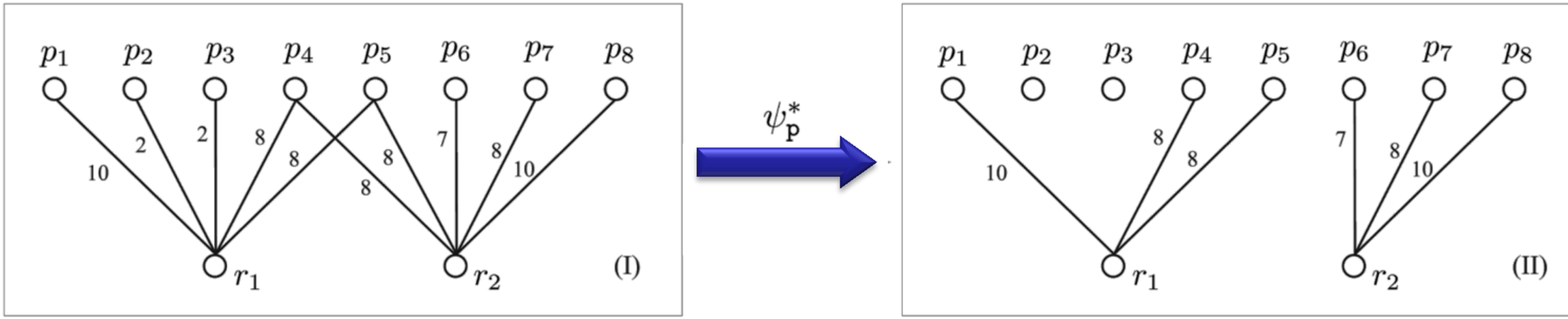


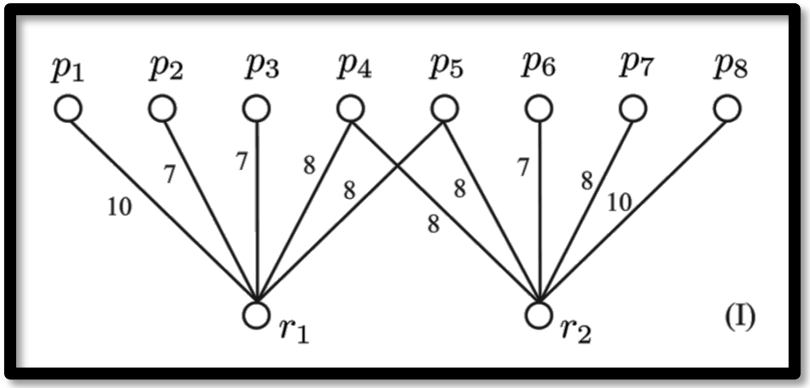
$$\mathtt{proj}_r(\psi^*) = \sum_{p \in \psi*(r)} score_{\mathrm{VQR}}(p)$$

Fairness Issues: `proj`

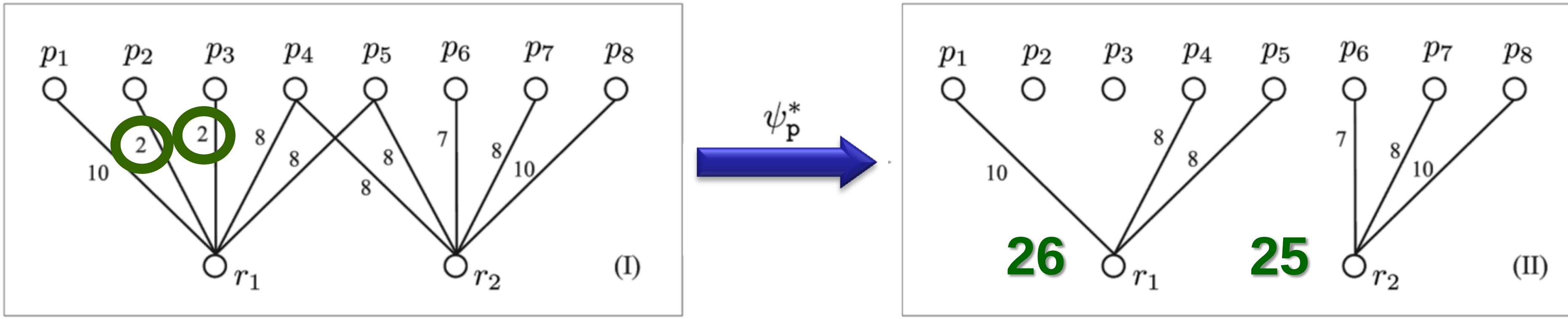


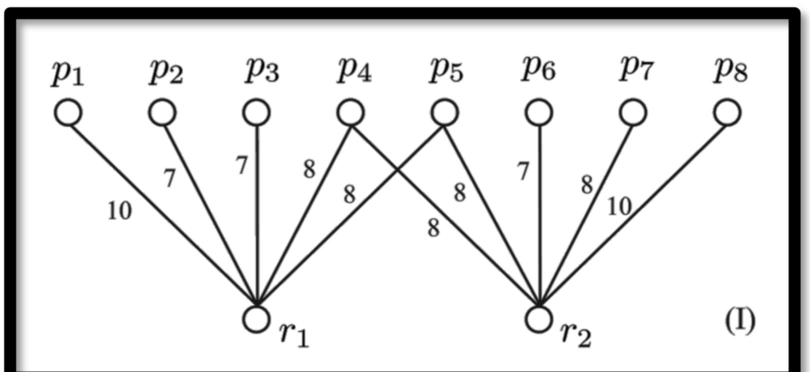
Strategic Manipulations: `proj`





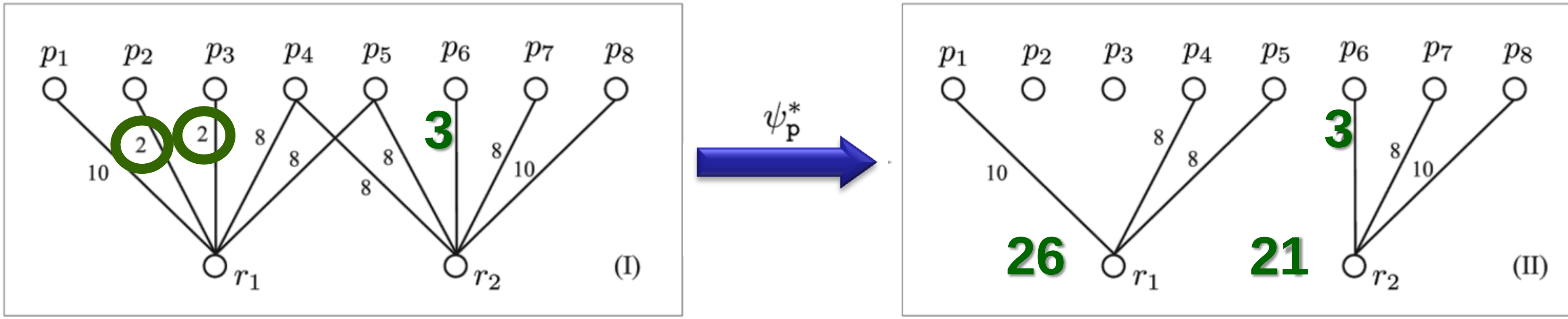
Strategic Manipulations: `proj`

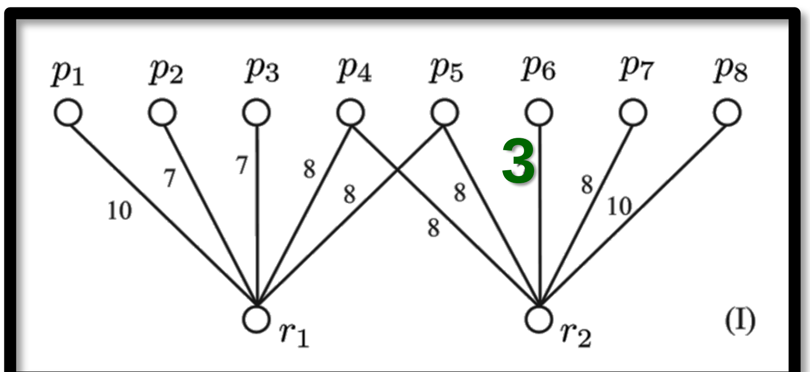




Under- estimation

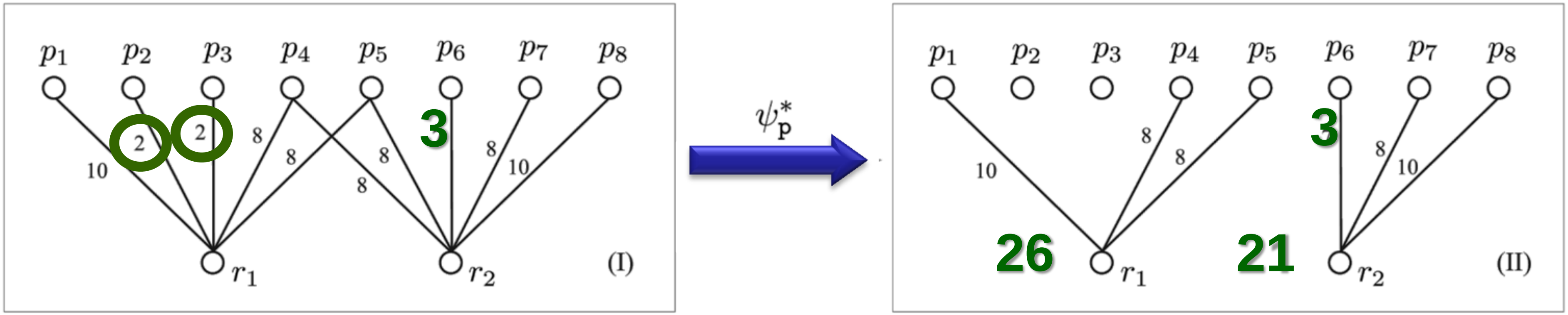
Strategic Manipulations: `proj`

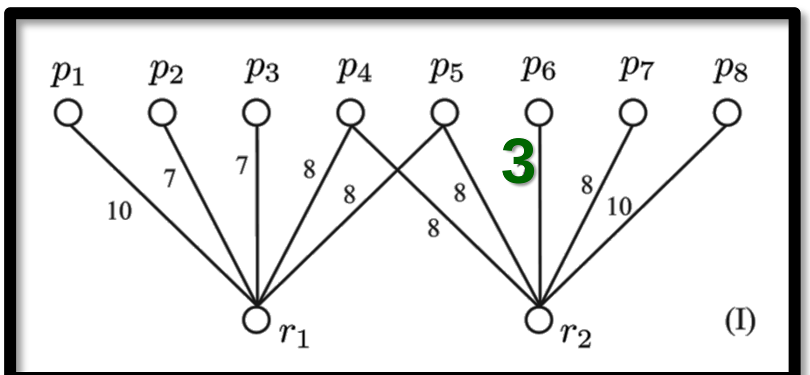




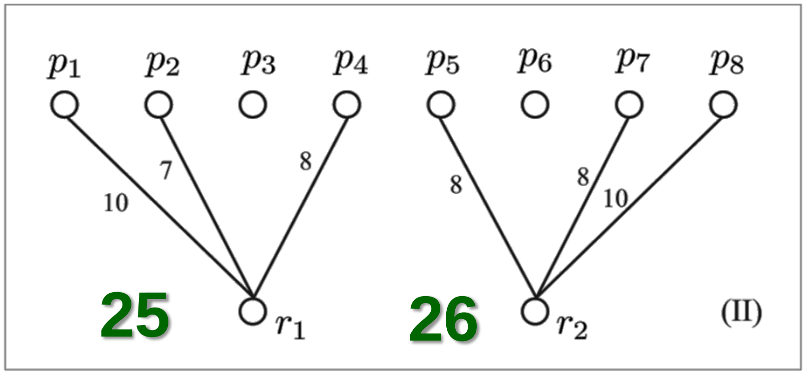
Even worse…

Strategic Manipulations: `proj`



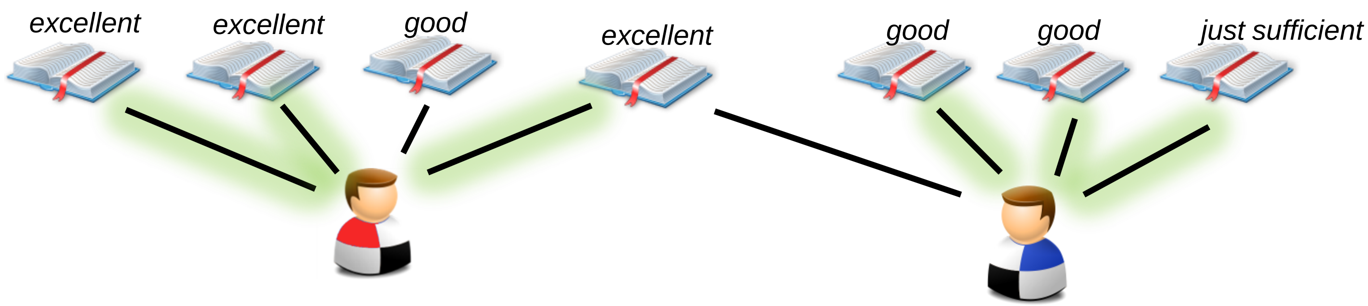


Even worse…



From Theory to ANVUR

- ANVUR did not specify a division rule
- Reserchers considered `proj` as «the rule»
- Researchers submitted (rated) only the minimum number of publications required (by default 3), thus implicitly under-estimating all their other products
- To avoid overlapping submissions, «aggrements» have been made



- The allocation has already been done
- Strategic manipulations happened
- Universities hardly found the optimal allocation

Outline

- The Model
- An Application Scenario
- **Algorithms and Results**

The Mechanism...

Input:	An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption:	A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

1. Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
2. For each set $\mathcal{C} \in \mathbb{C}$,
3. $\lfloor$ Compute an optimal allocation $\pi_{\mathcal{C}}$ for $\langle \mathcal{C}, \texttt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. $|$ For each set $\mathcal{C} \in \mathbb{C}$,
6. $|$ $|$ Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $\quad (= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. $|$ $\lfloor$ Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $\quad (= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
8. $|$ Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}|-1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
9. $\lfloor$ Define $p^{\xi}_i(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

The Mechanism…

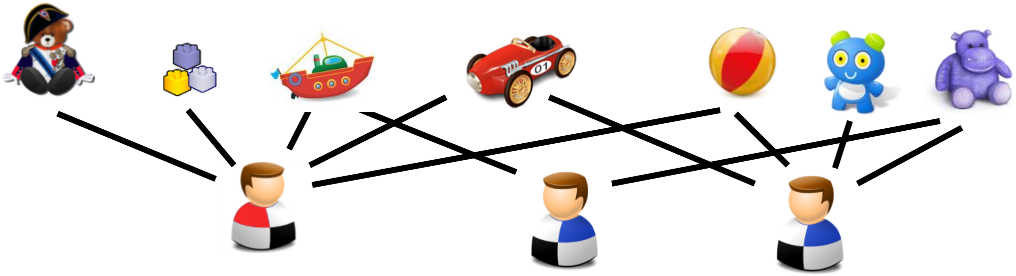
Input: An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption: A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

---

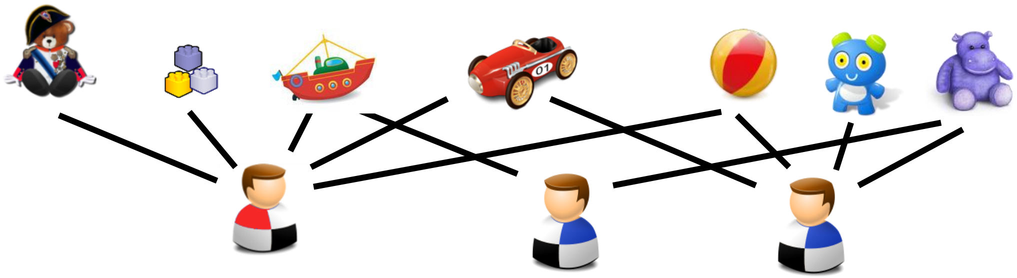
1. Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
2. For each set $\mathcal{C} \in \mathbb{C}$,
3. $\quad$ Compute an optimal allocation $\pi_{\mathcal{C}}$ for $\langle \mathcal{C}, \mathtt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. $\quad$ For each set $\mathcal{C} \in \mathbb{C}$,
6. $\quad\quad$ Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $\quad (= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. $\quad\quad$ Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $\quad (= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
8. $\quad$ Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}|-1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
9. $\quad$ Define $p^{\xi}_i(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

Allocated goods are considered only

A Key Lemma

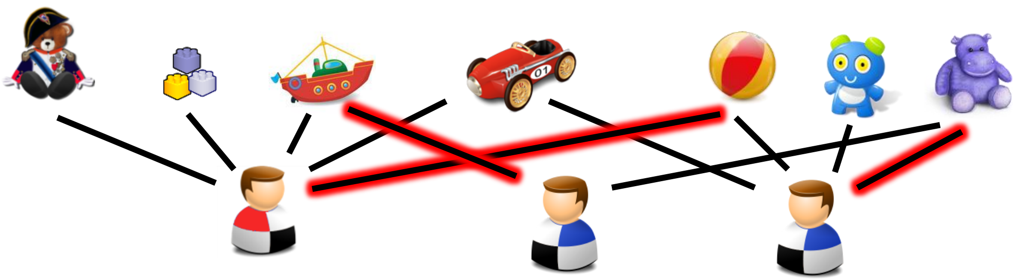


A Key Lemma



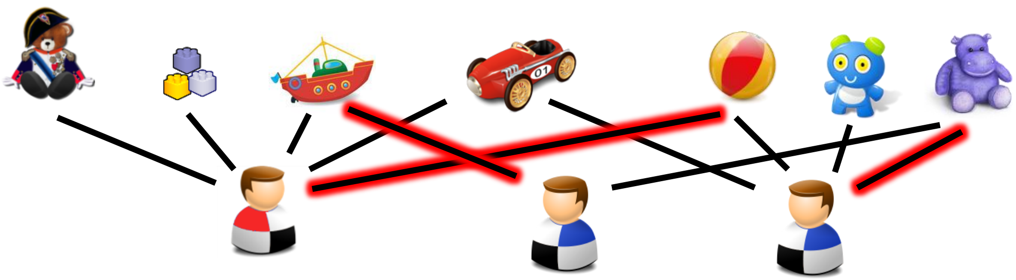
- Consider an optimal allocation (w.r.t. some declared types)

A Key Lemma



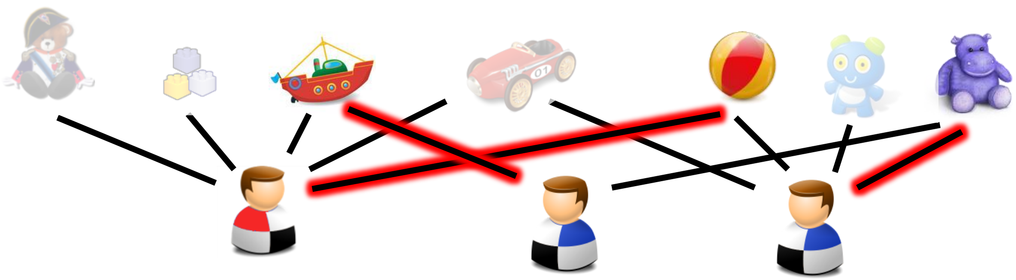
- Consider an optimal allocation (w.r.t. some declared types)

A Key Lemma



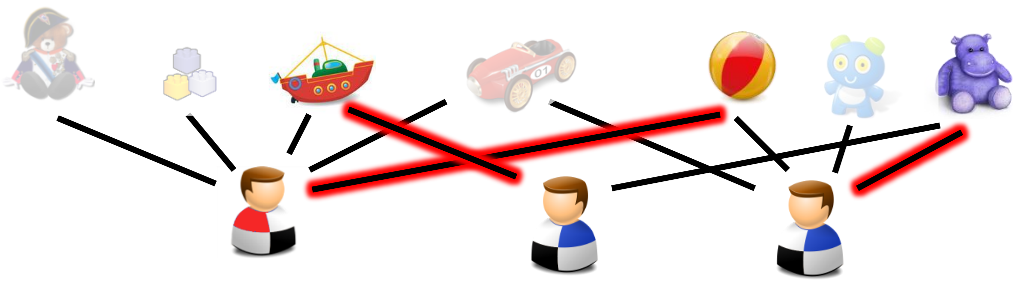
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…

A Key Lemma



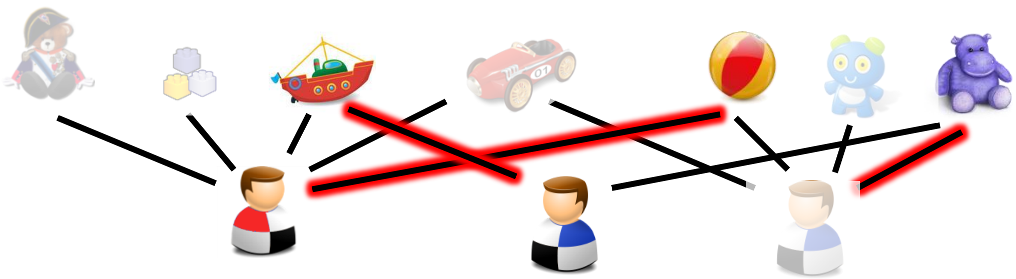
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…

A Key Lemma



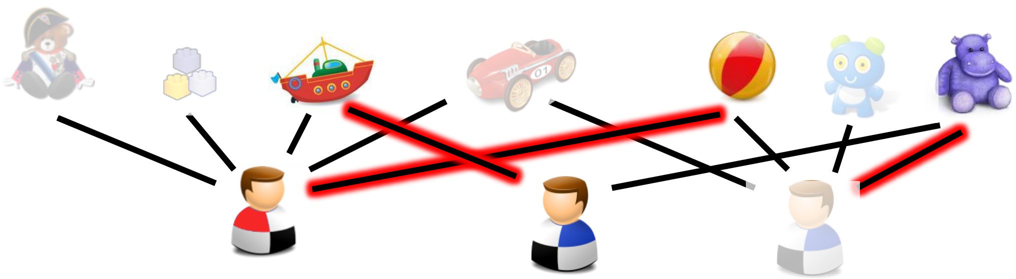
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents

A Key Lemma



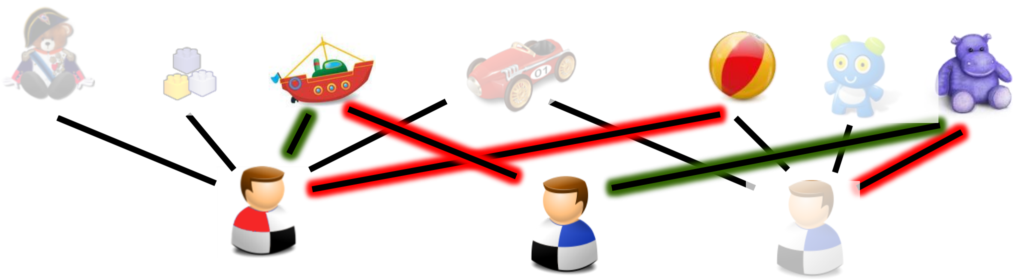
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents

A Key Lemma



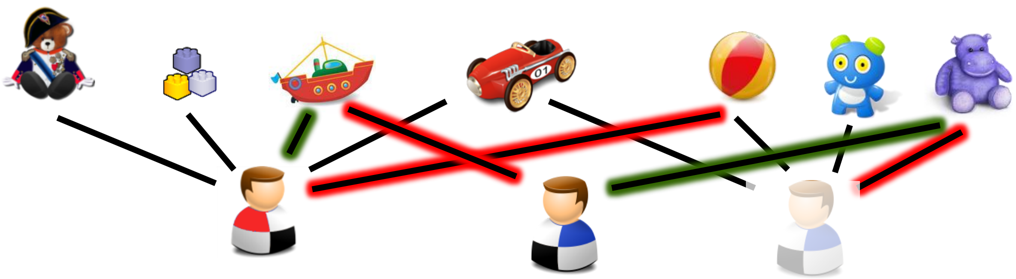
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents
- In this novel setting, compute an optimal allocation

A Key Lemma



- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents
- In this novel setting, compute an optimal allocation

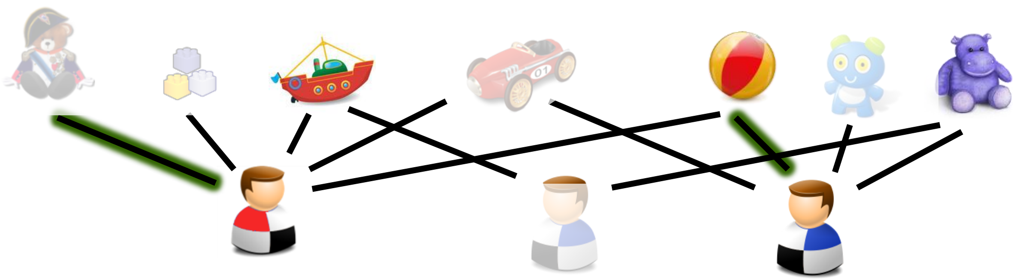
A Key Lemma

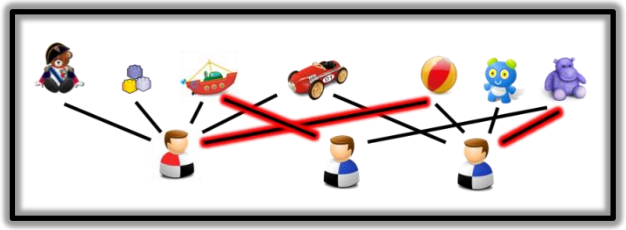


- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents
- In this novel setting, compute an optimal allocation

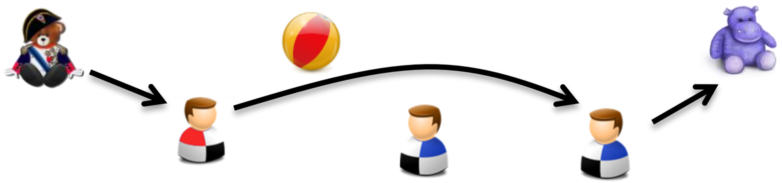
❖ **The allocation is also optimal for that coalition, even if all goods were actually available**

A Key Lemma





Notion of “update graph” with “flow” arguments



❖ **The allocation is also optimal for that coalition, even if all goods were actually available**

The Mechanism…

Input: An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption: A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

---

1. Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
2. For each set $\mathcal{C} \in \mathbb{C}$,
3. $\lfloor$ Compute an optimal allocation $\pi_{\mathcal{C}}$ for $\langle \mathcal{C}, \texttt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. $|$ For each set $\mathcal{C} \in \mathbb{C}$,
6. $|$ $|$ Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $\quad (= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. $|$ $\lfloor$ Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $\quad (= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
8. $|$ Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}|-1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
9. $\lfloor$ Define $p^{\xi}_i(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

Allocated goods are considered only

By the previous lemma, this is without loss of generality
In fact, allocated goods are the only ones that we verify

The Mechanism...

Input: An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption: A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

---

1. Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
2. For each set $\mathcal{C} \in \mathbb{C}$,
3. ⌊ Compute an optimal allocation $\pi_{\mathcal{C}}$ for $\langle \mathcal{C}, \mathtt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. | For each set $\mathcal{C} \in \mathbb{C}$,
6. | | Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $\quad (= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. | ⌊ Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $\quad (= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
8. | Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}|-1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
9. ⌊ Define $p^{\xi}_i(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

Allocated goods are considered only

«Bonus and Compensation»,
by Nisan and Ronen (2001)

The Mechanism…

Input: An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption: A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
For each set $\mathcal{C} \in \mathbb{C}$,
  Compute an optimal allocation $\pi_{\mathcal{C}}$ for $\langle \mathcal{C}, \mathtt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
For each agent $i \in \mathcal{A}$,
  For each set $\mathcal{C} \in \mathbb{C}$,
    Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
    Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
  Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}| - 1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
  Define $p^{\xi}_i(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

Allocated goods are considered only

«Bonus and Compensation», by Nisan and Ronen (2001)

No punishments!

The Mechanism…

Input: An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption: A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

1. Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
2. For each set $\mathcal{C} \in \mathbb{C}$,
3. Compute an optimal allocation $\pi_{\mathcal{C}}$ for $\langle \mathcal{C}, \mathtt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. For each set $\mathcal{C} \in \mathbb{C}$,
6. Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
8. Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}|-1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
9. Define $p_i^{\xi}(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

Allocated goods are considered only

«Bonus and Compensation», by Nisan and Ronen (2001)

❖ **Truth-telling is a dominant strategy for each agent**

The Mechanism…

Input: An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption: A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

1. Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
2. For each set $\mathcal{C} \in \mathbb{C}$,
3. Compute an optimal allocation $\pi_{\mathcal{C}}$ for $\langle \mathcal{C}, \mathtt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. For each set $\mathcal{C} \in \mathbb{C}$,
6. Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
8. Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}| - 1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
9. Define $p^{\xi}_i(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

Allocated goods are considered only

«Bonus and Compensation», by Nisan and Ronen (2001)

Does not depend on *i*

Is maximized when the declared type coincides with the verified one

❖ **Truth-telling is a dominant strategy for each agent**

The Mechanism…

Input: An allocation π for $\langle \mathcal{A}, G, \omega \rangle$, and a vector $\mathbf{w} \in \mathbf{D}$;
Assumption: A verifier $\mathbf{v}$ is available. Let $\mathbf{v}(\pi) = (v_1, \ldots, v_n)$;

---

1. Let $\mathbb{C}$ denote the set of all possible subsets of $\mathcal{A}$;
2. For each set $\mathcal{C} \in \mathbb{C}$,
3. | Compute an optimal allocation $\pi_\mathcal{C}$ for $\langle \mathcal{C}, \mathtt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. | For each set $\mathcal{C} \in \mathbb{C}$,
6. | | Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_\mathcal{C}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_\mathcal{C}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_\mathcal{C}))$;
7. | | Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
8. | Let $\xi_i(\pi, \mathbf{w}) := \sum_{\mathcal{C} \in \mathbb{C}} \frac{(|\mathcal{A}| - |\mathcal{C}|)!(|\mathcal{C}|-1)!}{|\mathcal{A}|!} (\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) - \Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}))$;
9. | Define $p_i^\xi(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

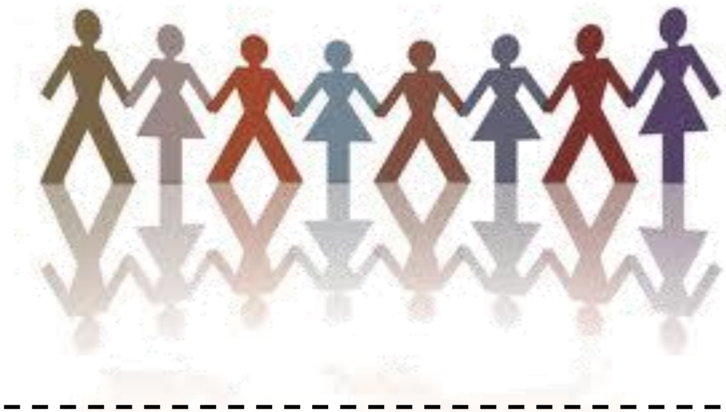
Allocated goods are considered only

«Bonus and Compensation», by Nisan and Ronen (2001)

❖ **Truth-telling is a dominant strategy for each agent**

Coalitional Games

- Players form *coalitions*
- Each coalition is associated with a *worth*
- A *total worth* has to be distributed



$$\mathcal{G} = \langle N, \varphi \rangle, \; \varphi : 2^N \mapsto \mathbb{R}$$

- **Solution Concepts** characterize outcomes in terms of
  - Fairness
  - Stability

Coalitional Games: Shapley Value

$$\phi_i(\mathcal{G}) = \sum_{C \subseteq N} \frac{(|N| - |C|)!(|C| - 1)!}{|N|!}(\varphi(C) - \varphi(C \setminus \{i\}))$$

- **Solution Concepts** characterize outcomes in terms of
  - Fairness
  - Stability

The Mechanism

$$\mathcal{G} = \langle N, \varphi \rangle, \ \varphi : 2^N \mapsto \mathbb{R}$$

- $\varphi(C)$ is the *contribution* of the coalition **w.r.t.** $\left\{ \begin{array}{l} \textbf{selected products} \\ \textit{and} \\ \textbf{verified values} \end{array} \right.$

The Mechanism

$$\mathcal{G} = \langle N, \varphi \rangle, \ \varphi : 2^N \mapsto \mathbb{R}$$

- $\varphi(C)$ is the *contribution* of the coalition **w.r.t.** { **selected products** *and* **verified values** }

Best possible allocation,
assuming that agents in C are the only ones in the game

The Mechanism

$\mathcal{G} = \langle N, \varphi \rangle$, $\varphi : 2^N \mapsto \mathbb{R}$

- $\varphi(C)$ is the *contribution* of the coalition **w.r.t.** { **selected products** *and* **verified values (π)**

Each researcher gets the Shapley value $\phi_i(\mathcal{G})$

The Mechanism

$\mathcal{G} = \langle N, \varphi \rangle$, $\varphi : 2^N \mapsto \mathbb{R}$

- $\varphi(C)$ is the *contribution* of the coalition **w.r.t.** { **selected products** *and* **verified values (**π**)** }

Each researcher gets the Shapley value $\phi_i(\mathcal{G})$

Properties

The resulting mechanism is «fair» and «buget balanced»

The Mechanism

$\mathcal{G} = \langle N, \varphi \rangle$, $\varphi : 2^N \mapsto \mathbb{R}$

- $\varphi(C)$ is the *contribution* of the coalition **w.r.t.** { **selected products** *and* **verified values (π)** }

Each researcher gets the Shapley value $\phi_i(\mathcal{G})$

Properties

The resulting mechanism is «fair» and «buget balanced»

$$\sum_{i \in N} \phi_i(\mathcal{G}) = \varphi(N)$$

Key Observations for Fairness

- Let π be an optimal allocation
- Let π' be an allocation

Key Observations for Fairness

- Let π be an optimal allocation
- Let π' be an allocation

$\mathcal{G} \leftarrow \pi$

$\varphi(C)$

(best allocation for the coalition with products in π)

As π is optimal, then $\varphi(C)$ is in fact optimal even by considering all possible products as available

π'

$\mathcal{G}'$

$\varphi(C) \geq \varphi'(C)$

Key Observations for Fairness

- Let π be an optimal allocation
- Let π' be an allocation

$$\mathcal{G} \longleftarrow \pi$$
$$\downarrow$$
$$\varphi(C)$$

(best allocation for the coalition with products in π)

As π is optimal, ther $\varphi(C)$ is in fact optimal even by considering all possible products as available

$$\pi'$$
$$\downarrow$$
$$\mathcal{G}'$$
$$\downarrow$$
$$\varphi(C) \geq \varphi'(C)$$

By the monotonicity of the Shapley value, $\phi_i \geq \phi_i'$

Key Observations for Fairness

- Let π be an optimal allocation
- Let π' be an allocation

$$\pi \geq \pi'$$

❖ **Optimal allocations are always preferred**
❖ **There is no difference between two different optimal allocations**

Complexity Issues

- For many classes of «compact games» (e.g., *graph games*), the Shapley-value can be efficiently calculated
- Here, the problem emerges to be #P-complete

Complexity Issues

- For many classes of «compact games» (e.g., *graph games*), the Shapley-value can be efficiently calculated
- Here, the problem emerges to be #P-complete

- #P is the class the class of all functions that can be computed by *counting Turing machines* in polynomial time
- A counting Turing machine is a standard nondeterministic Turing machine with an auxiliary output device that prints in binary notation the number of accepting computations induced by the input
- Prototypical problem: to count the number of truth variable assignments that satisfy a Boolean formula

Complexity Issues

- For many classes of «compact games» (e.g., *graph games*), the Shapley-value can be efficiently calculated
- Here, the problem emerges to be #P-complete

Reduction from the problem of counting the number of perfect matchings in certain bipartite graphs [Valiant, 1979]

- #P is the class the class of all functions that can be computed by *counting Turing machines* in polynomial time
- A counting Turing machine is a standard nondeterministic Turing machine with an auxiliary output device that prints in binary notation the number of accepting computations induced by the input
- Prototypical problem: to count the number of truth variable assignments that satisfy a Boolean formula

Complexity Issues

- #P-complete
- Practically feasible, if used for substructures
- Moreover…

Complexity Issues

- #P-complete
- Practically feasible, if used for substructures
- Moreover…

Fully Polynomial-Time Randomized Approximation Scheme

- Always Efficient and Budget Balanced
- All other properties in expectation (with high probability)

Coupling of the algorithm with a sampling strategy for the coalitions by [Liben-Nowell,Sharp, Wexler, Woods; 2012]

Outline

The Model

An Application Scenario

Algorithms and Results

Outline

The Model

An Application Scenario

Algorithms and Results

Back to ANVUR

Recent Good News

- Many collegues (not just computer scientists) have now recognized the problem
- In fact while recognizing the strategic issues is «not easy», there is still the problem to ditribute the score of each University among the Departments…

Recent Good News

- Many collegues (not just computer scientists) have now recognized the problem
- In fact while recognizing the strategic issues is «not easy», there is still the problem to ditribute the score of each University among the Departments…

- Our method has been presented to the ANVUR President
- The Shapley-value approach can still be used, even if (of course) efficiency cannot be guaranteed
- At least, it provides a fair appoach to worth distribution

Recent Good News

- There are chances for its adoption as the method for worth distribution over Departments, and the whole approach might be used in the next evaluation
- Implementation with suitable data structures and methods to speed-up the computation at «La Sapienza», Rome.

[Schaerf, et al.]

- Our method has been presented to the ANVUR President
- The Shapley-value approach can still be used, even if (of course) efficiency cannot be guaranteed
- At least, it provides a fair appoach to worth distribution

Thank you!