17th Italian Conference on Theoretical Computer Science
Lecce, Italy, September 7-9th 2016

Mechanisms with Verification and Fair Allocation Problems

Gianluigi Greco

UNIVERSITÀ DELLA CALABRIA

Based on:
- Mechanisms for Fair Allocation Problems. JAIR 2014
- Structural Tractability of Shapley and Banzhaf Values in Allocation Games. IJCAI 2015
- Fair division rules for funds distribution. Intelligenza Artificiale 2013

See also:
- The Complexity of the Nucleolus in Compact Games. TOCT 2014
- Hypertree Decompositions: Questions and Answers. PODS 2016

Outline

- Background on Mechanism Design
- Mechanisms for Allocation Problems
- Complexity Analysis
- Case Study

Social Choice Functions

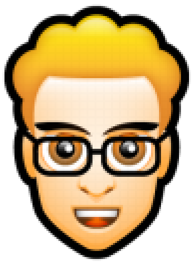
Rule for breaking ties: A > B > C

Alternatives

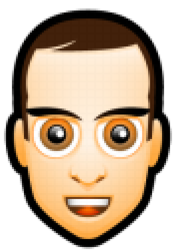
- { A, B, C }

Social Choice Function:

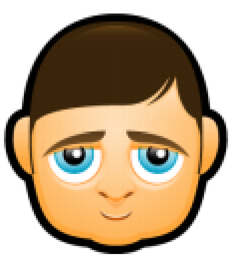
- Compute the alternative that is top-ranked by the majority



A > C > B



B > A > C



C > B > A

A

Social Choice Functions

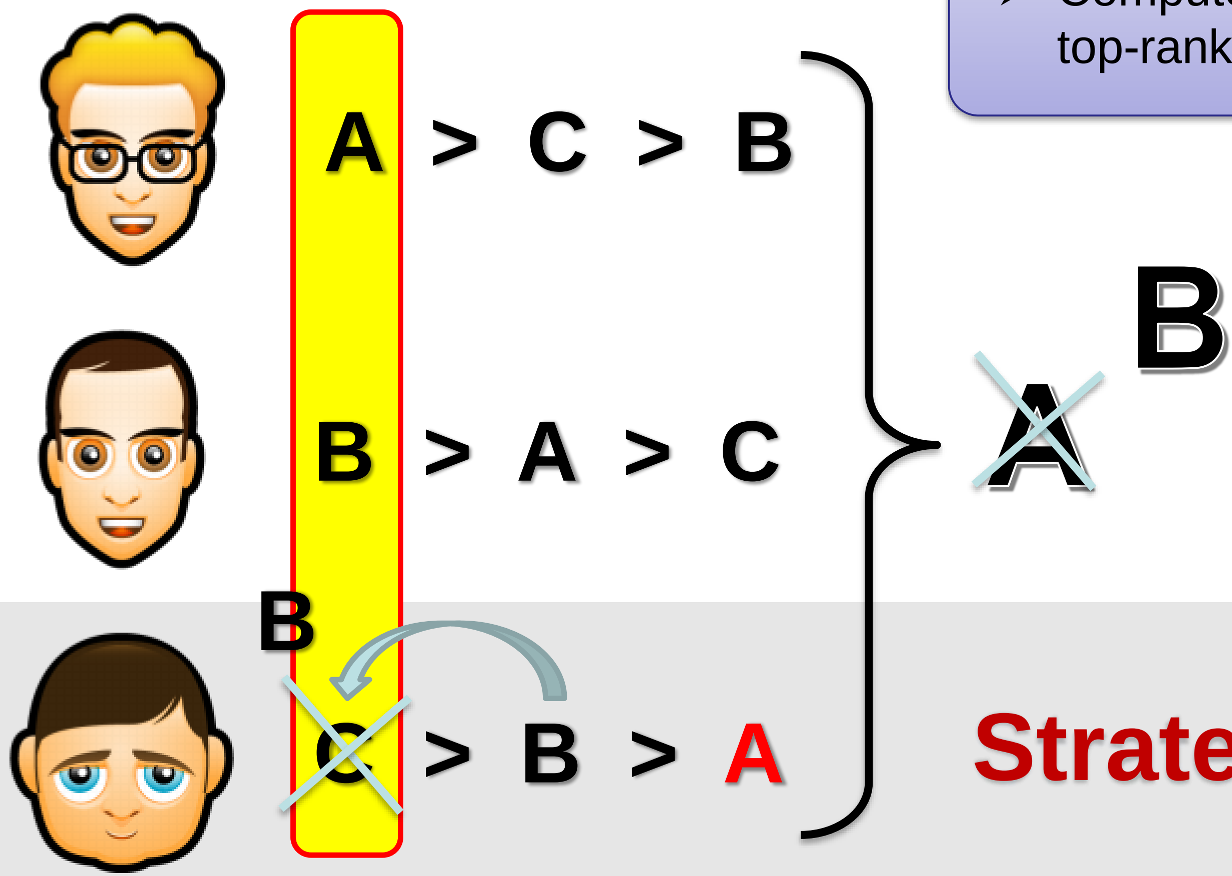
Rule for breaking ties: A > B > C

Alternatives

- { A, B, C }

***Social Choice Function*:**

- Compute the alternative that is top-ranked by the majority



Strategic issues!

Mechanism Design

- Social Choice Theory is *non-strategic*
- In practice, agents **declare** their preferences
  - They are self interested
  - They might not reveal their true preferences
- We want to find optimal outcomes w.r.t. true preferences
- Optimizing w.r.t. the declared preferences might not achieve the goal

How to build a mechanism where agents find convenient to report their true preferences?

Basic Concepts (1/2)

- Each agent i is associated with a **type** $\theta_i \in \Theta_i$

private knowledge, preferences,…

C > B > A

Basic Concepts (2/2)

- Consider the vector of the **joint strategies** $s = (s_1, \ldots, s_I)$

Solution Concepts

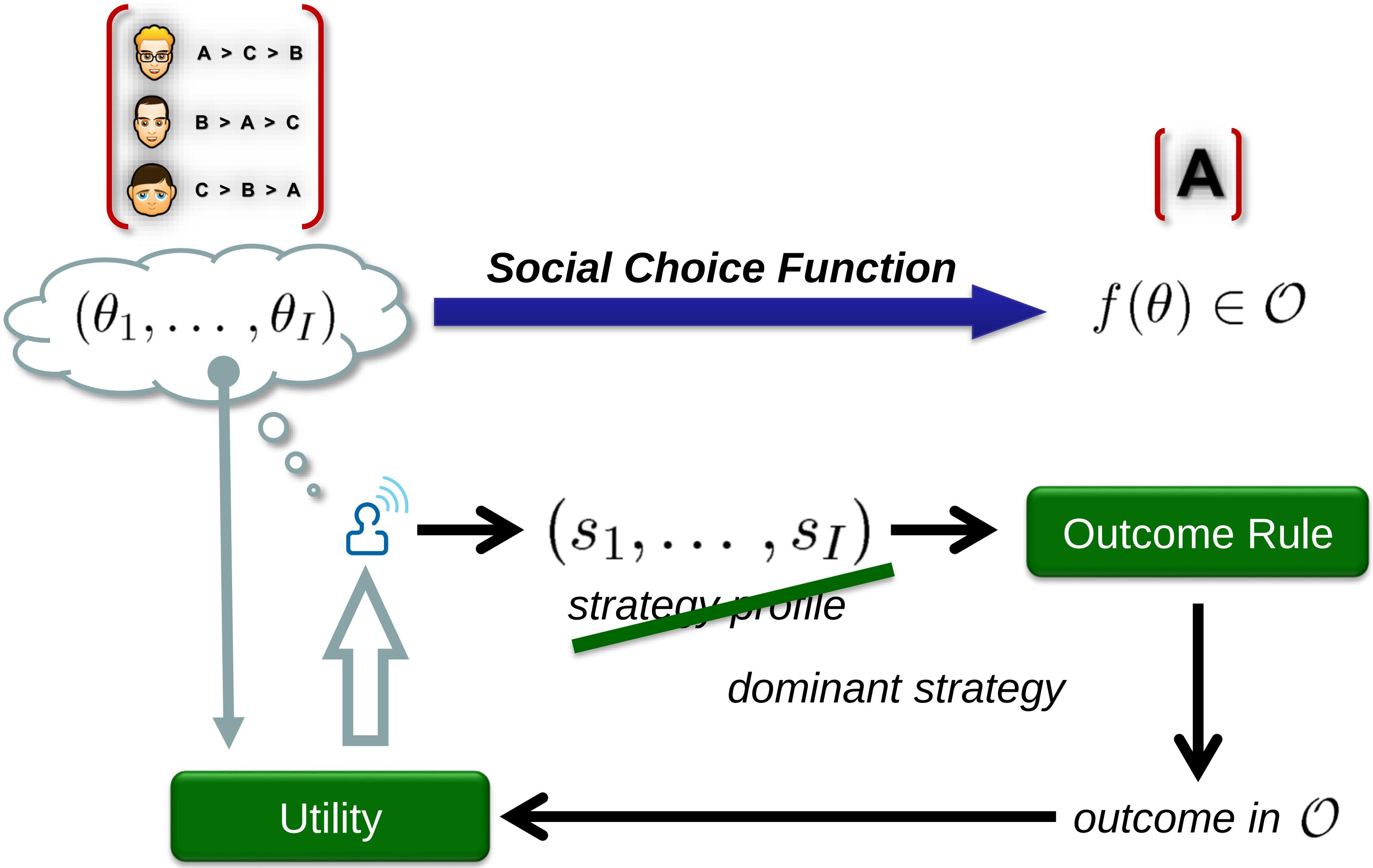
- A strategy s_i is **dominant** for agent i, if for every $s'_i \neq s_i$

  and for every s_{-i},

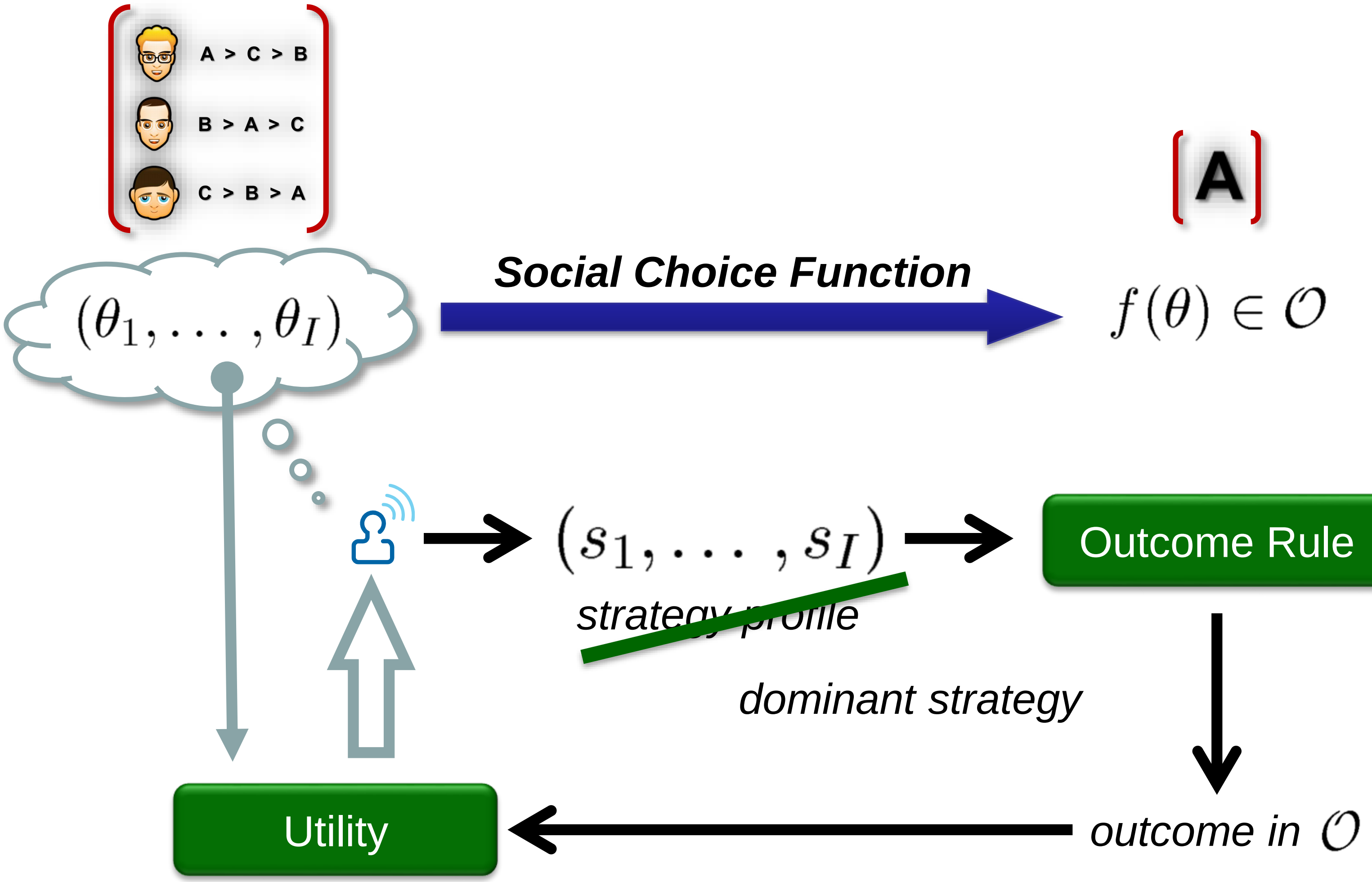
$$u_i(s_i, s_{-i}, \theta_i) \geq u_i(s'_i, s_{-i}, \theta_i)$$

Independently on the other agents…

Mechanism Design



Mechanism Design



- **The ideal goal is to build an outcome rule such that truth-telling is a dominant strategy**

Impossibility Result

- A social choice function is **dictatorial** if one agent always receives one of its most preferred alternatives

A > C > B

B > A > C

C > B > A

Which functions can be implemented in dominant strategies?

Impossibility Result

- A social choice function is **dictatorial** if one agent always receives one of its most preferred alternatives

THEOREM. Assume general preferences, at least two agents, and at least three optimal outcomes. A social choice function can be **implemented in dominant strategies** if, and only if, it is **dictatorial**.

- Very bad news...
  - [Gibbard, 1973] and [Satterthwaite, 1975]
- …, but must be interpreted with care

Which functions can be implemented in dominant strategies?

Payments

- Monetary compensation to induce **truthfulness**

- A utility is **quasi-linear** if it has the following form

$$u_i(o, \theta_i) = v_i(o, \theta_i) - p_i$$

valuation function
cardinal preferences

payment by the agent

Payments

- Monetary compensation to induce **truthfulness**

- A utility is **quasi-linear** if it has the following form

$$u_i(o, \theta_i) = v_i(o, \theta_i) - p_i$$

valuation function
cardinal preferences

payment by the agent

- Payments are defined by the mechanism

Payments and Desiderata

- Monetary compensation to induce **truthfulness**

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

Payments and Desiderata

- Monetary compensation to induce **truthfulness**

GOAL: Budget Balance

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

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

Payments and Desiderata

- Monetary compensation to induce **truthfulness**

GOAL: Budget Balance

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

- Monetary compensation to induce **fairness**
  - ✓ For instance, it is desirable that ***no agent envies*** the allocation of any another agent, or that
  - ✓ The outcome is ***Pareto efficient***, i.e., there is no different allocation such that every agent gets at least the same utility and one of them improves.

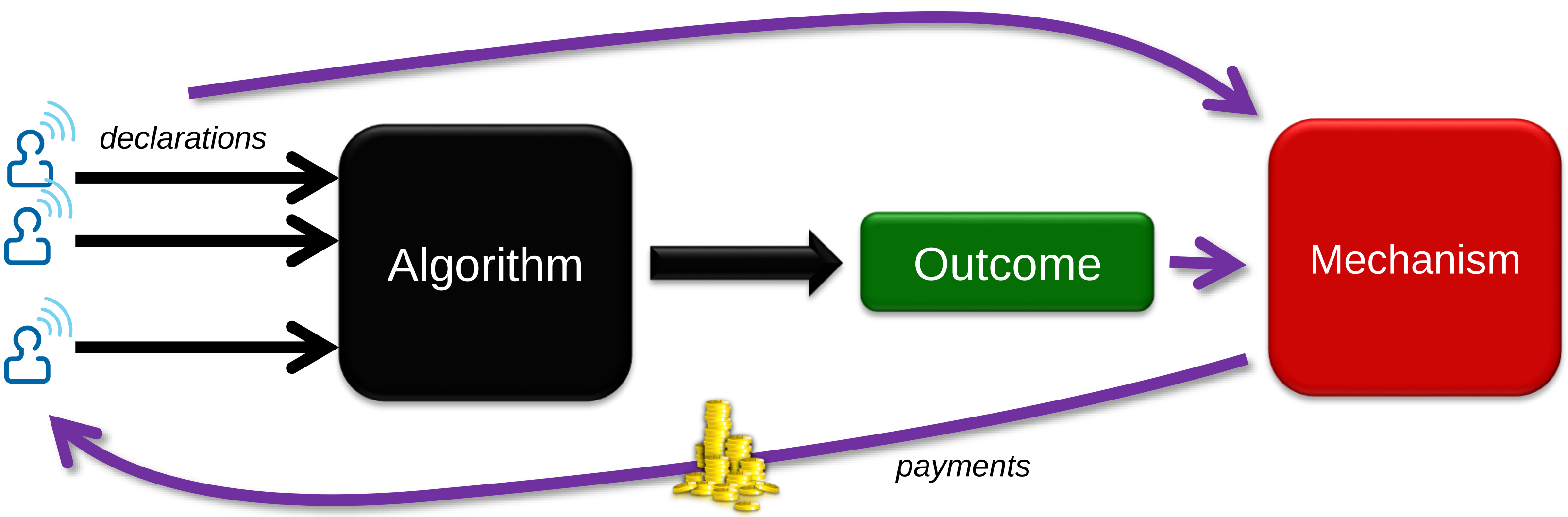
(A Few…) Impossibility Results

Efficiency + Truthfulness + Budget Balance

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

Fairness + Truthfulness + Budget Balance

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

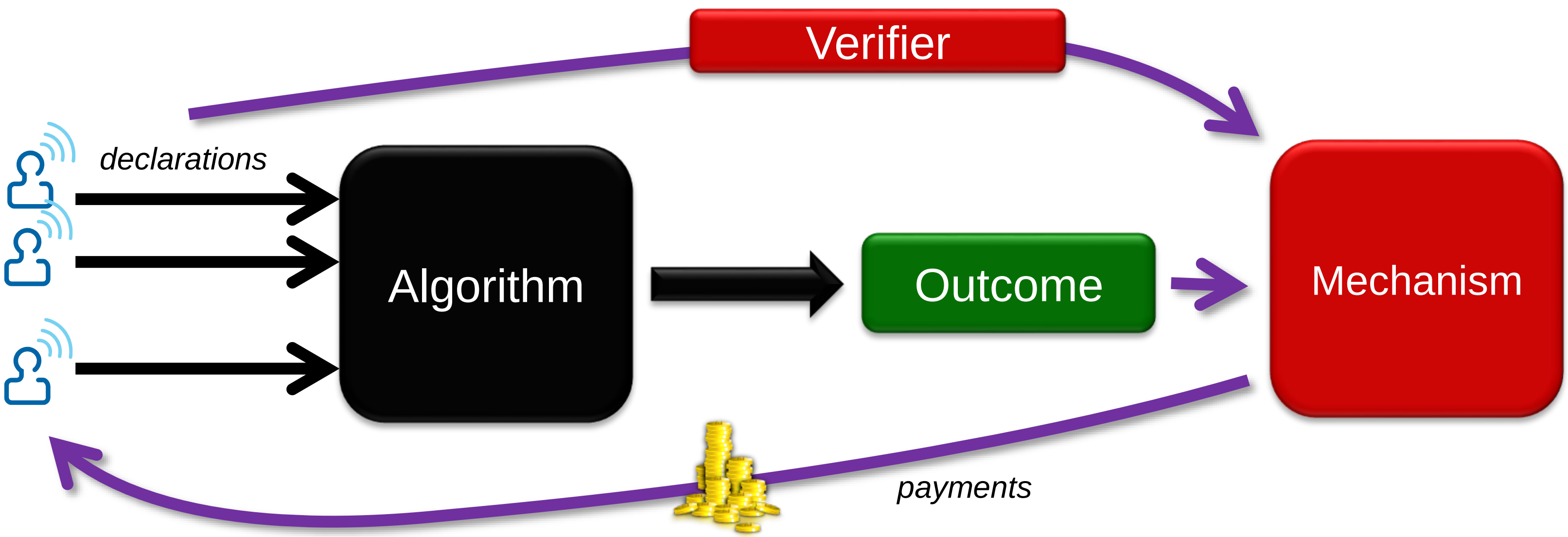


(A Few...) Impossibility Results

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

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

- Verification on «selected» declarations



Approaches to Verification

(1) Partial Verification

(2) Probabilistic Verification

Approaches to Verification

(1) Partial Verification

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

(2) Probabilistic Verification

Approaches to Verification

(1) Partial Verification

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

(2) Probabilistic Verification

Approaches to Verification

(1) Partial Verification

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

(2) Probabilistic Verification

[Caragiannis, Elkind, Szegedy, Yu; 2012]

Approaches to Verification

(1) Partial Verification

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

Approaches to Verification

(1) Partial Verification

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

- Verification is performed via **sensing**
  - Hence, it is subject to errors; for instance, because of the limited precision of the measurement instruments.
  - It might be problematic to decide whether an observed discrepancy between verified values and declared ones is due to a strategic behavior or to such sensing errors.

[Greco, Scarcello; 2014]

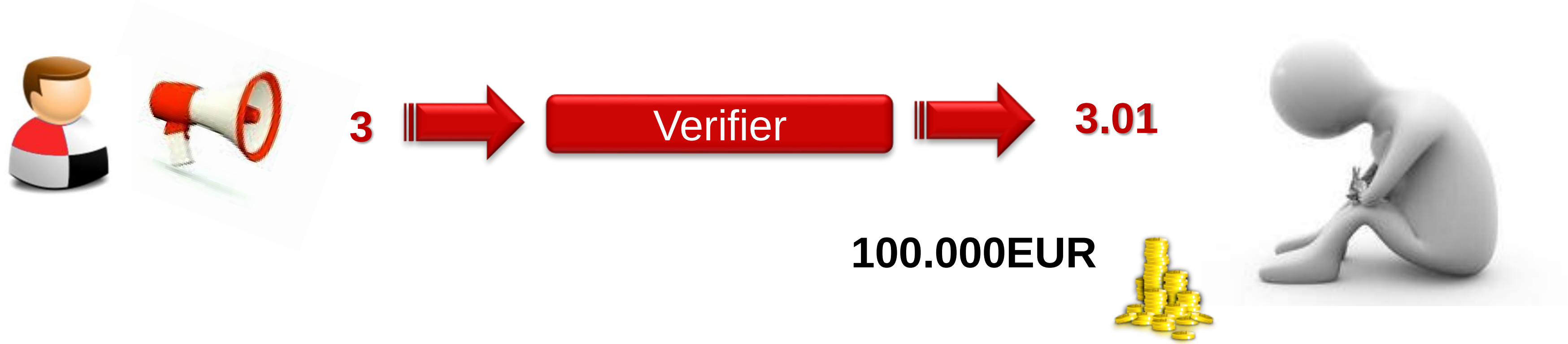
Approaches to Verification

- Verification is performed via **sensing**
  - Hence, it is subject to errors; for instance, because of the limited precision of the measurement instruments.
  - It might be problematic to decide whether an observed discrepancy between verified values and declared ones is due to a strategic behavior or to such sensing errors.

Approaches to Verification (bis)

- Agents might be uncertain of their private features; for instance, due to limited computational resources
  - There might be no strategic issues

Approaches to Verification (ter)



- Punishments enforce truthfulness
  - They might be disproportional to the harm done by misreporting
  - Inappropriate in real life situations in which uncertainty is inherent due to measurements errors or uncertain inputs.

[Feige, Tennenholtz; 2011]

Approaches to Verification

(1) Partial Verification

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

(3) Full Verification

The verifier returns a value.

Approaches to Verification

(1) Partial Verification

(2) Probabilistic Verification

Punishments are used to enforce truthfulness

(3) Full Verification

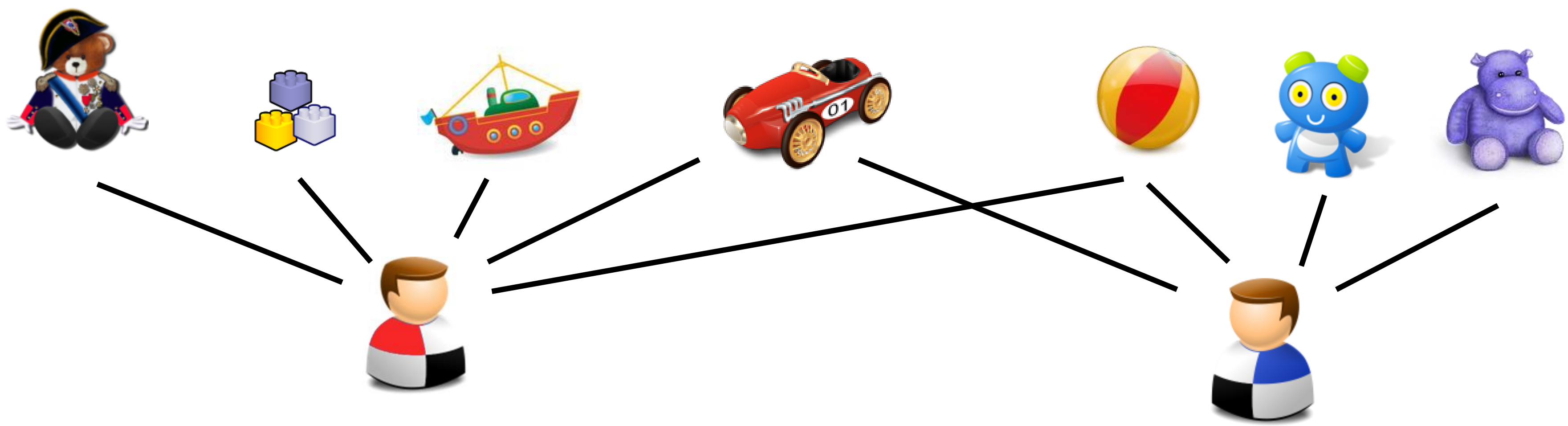
The verifier returns a value. But,…

- ***no punishment***
  - payments are always computed under the presumption of innocence, where incorrect declared values do not mean manipulation attempts by the agents
- **error tolerance**
  - the consequences of errors in the declarations produce a linear “distorting effect” on the various properties of the mechanism

Outline

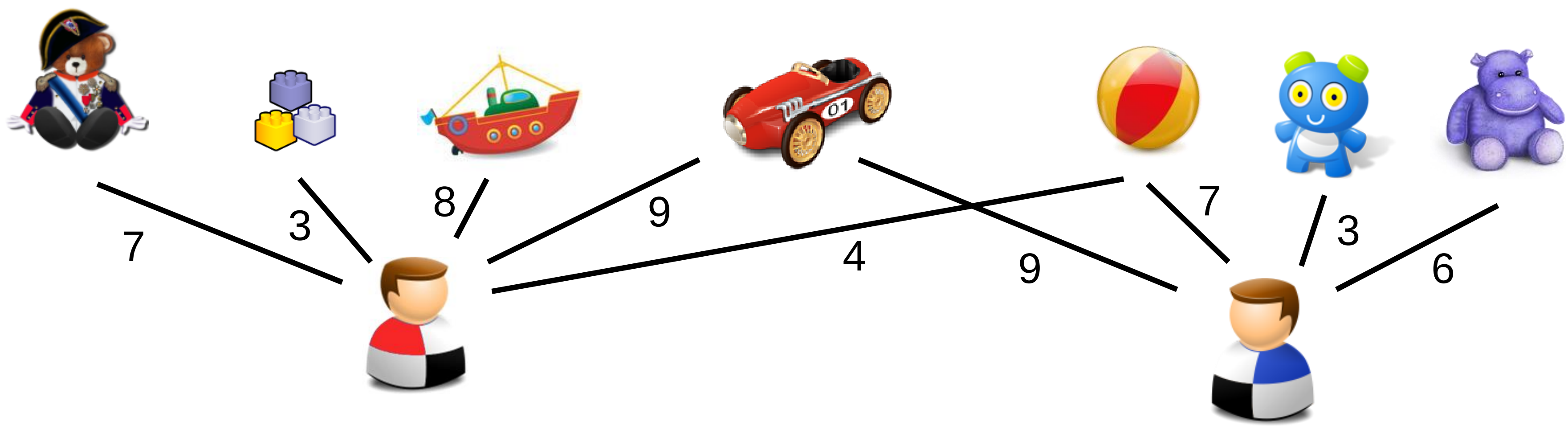
- Background on Mechanism Design
- Mechanisms for Allocation Problems
- Complexity Analysis
- Case Study

The Model



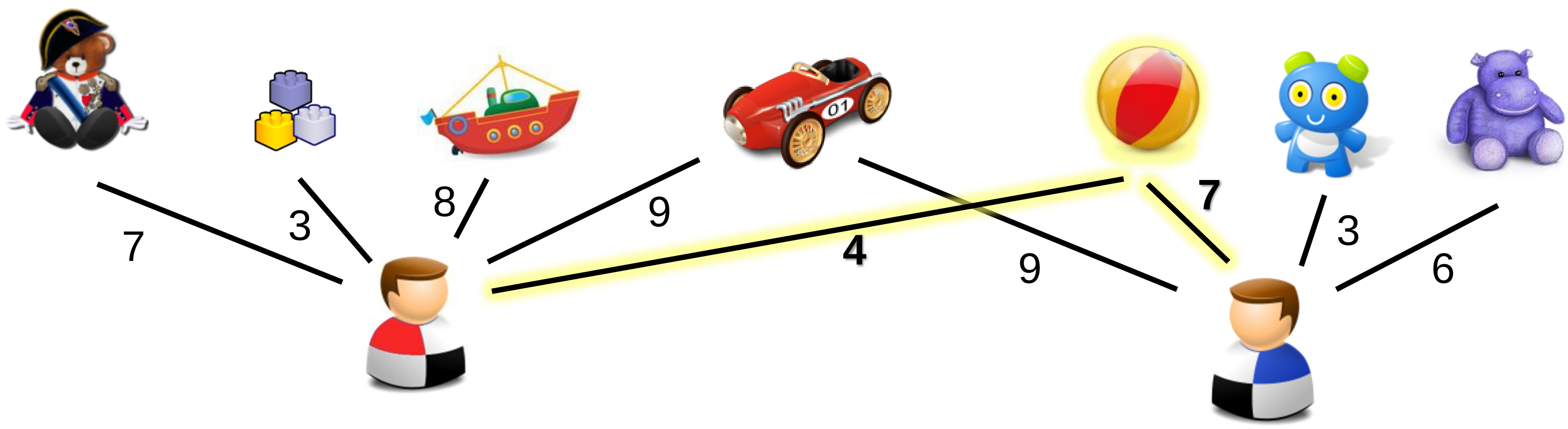
- Goods are indivisible and non-sharable
- Constraints on the maximum number of goods to be allocated to each agent
- Cardinal preferences: *Utility functions*

The Model



- Goods are indivisible and non-sharable
- Constraints on the maximum number of goods to be allocated to each agent
- Cardinal preferences: *Utility functions*

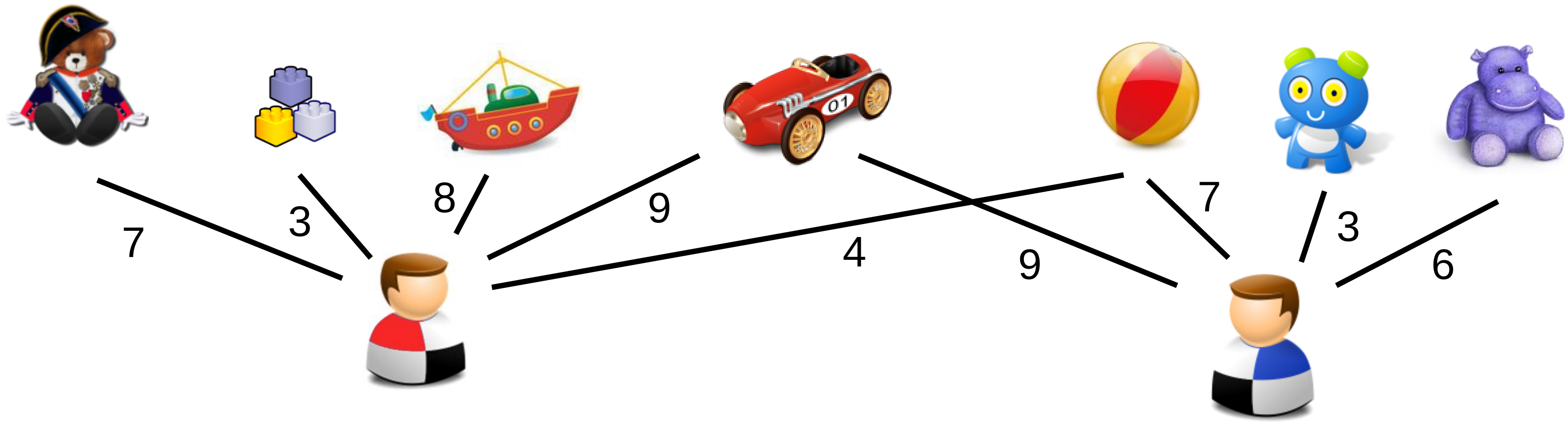
The Model



- Goods are indivisible and non-sharable
- Constraints on the maximum number of goods to be allocated to each agent
- Cardinal preferences: *Utility functions*

Different agents might have different valuations for the same good

The Model

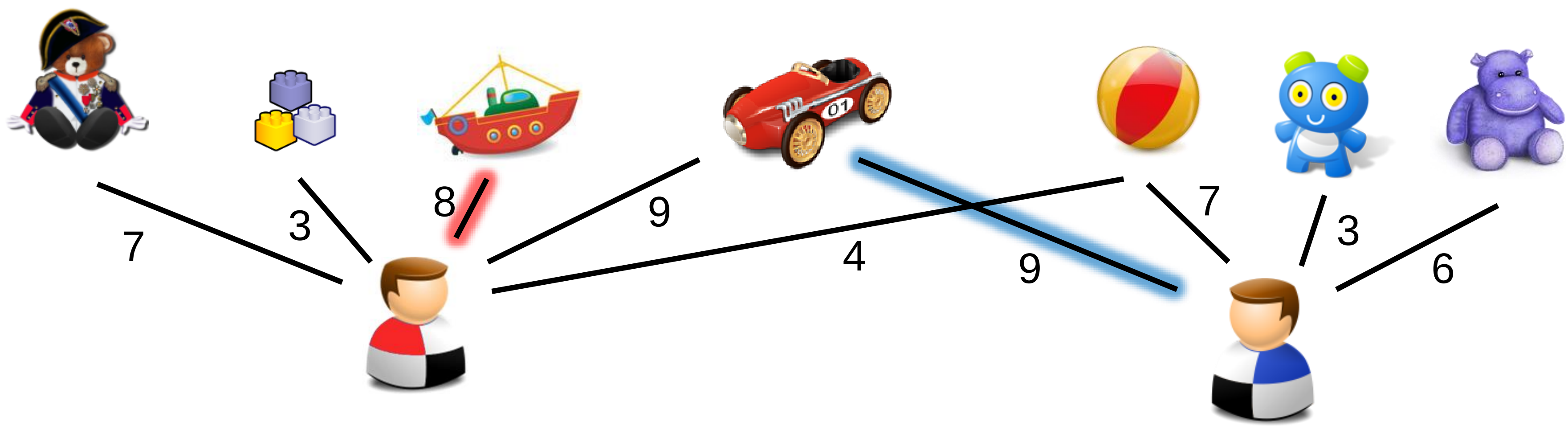


- Goods are indivisible and non-sharable
- Constraints on the maximum number of goods to be allocated to each agent
- Cardinal preferences: *Utility functions*

GOAL: Optimal Allocations

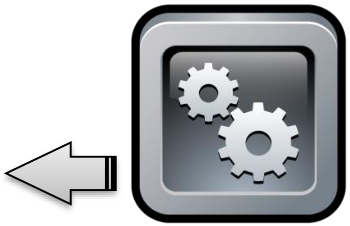
✓ Social Welfare
✓ Efficiency

The Model



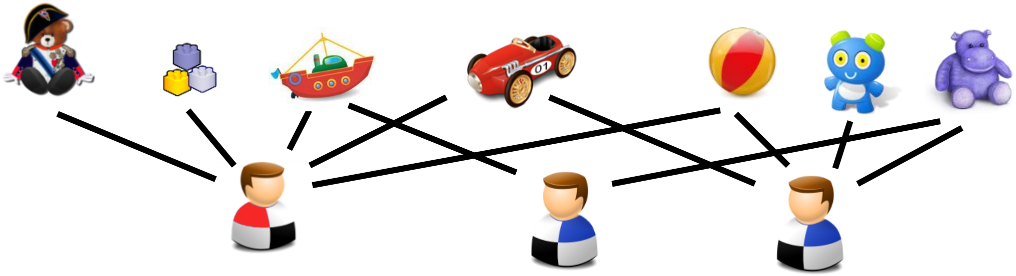
- Goods are indivisible and non-sharable
- Constraints on the maximum number of goods to be allocated to each agent
- Cardinal preferences: *Utility functions*

GOAL: Optimal Allocations

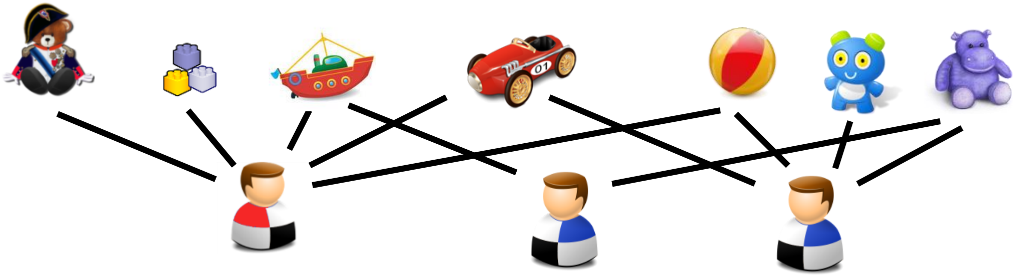


✓ Social Welfare
✓ Efficiency

A Key Lemma

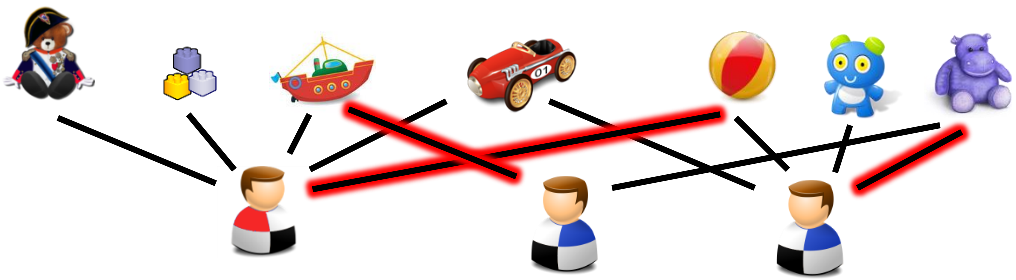


A Key Lemma



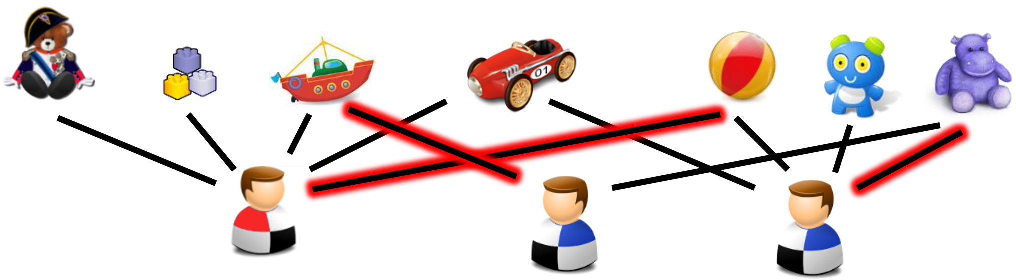
- Consider an optimal allocation (w.r.t. some declared types)

A Key Lemma



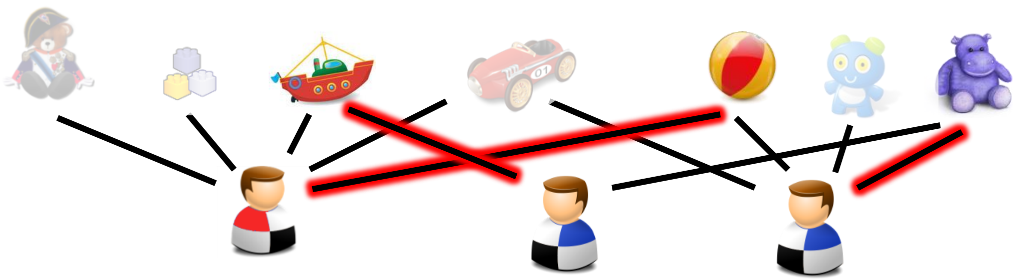
- Consider an optimal allocation (w.r.t. some declared types)

A Key Lemma



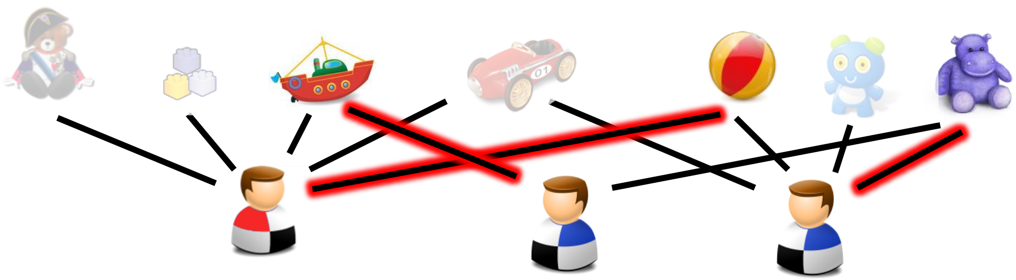
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…

A Key Lemma



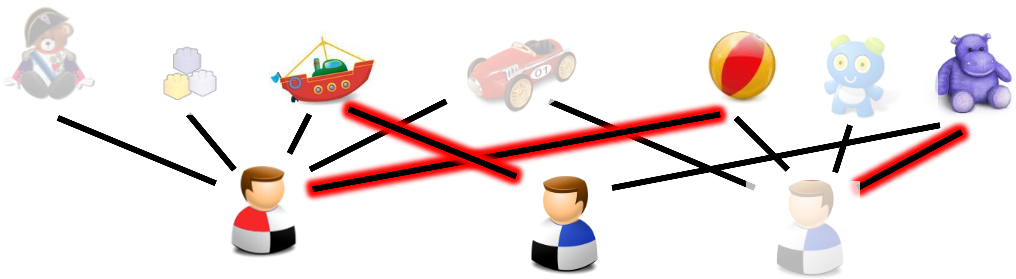
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…

A Key Lemma



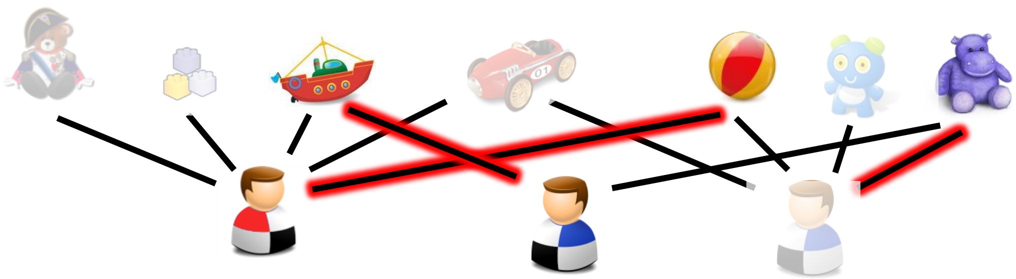
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents

A Key Lemma



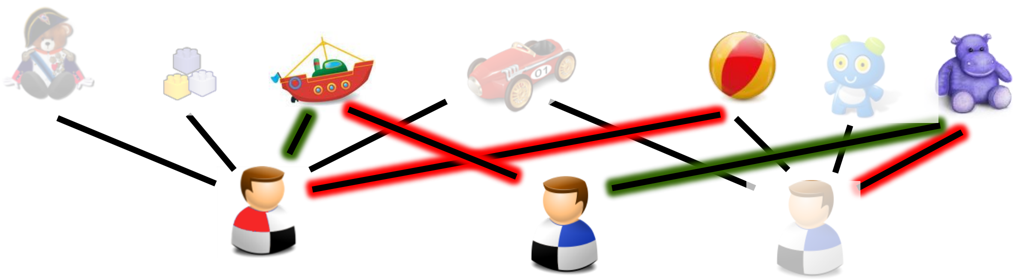
- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents

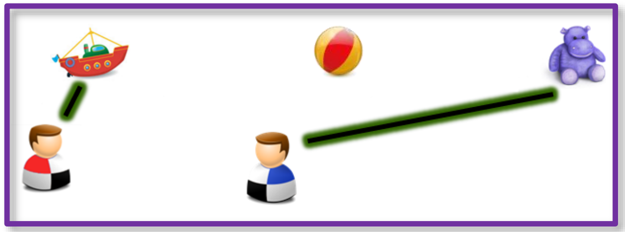
A Key Lemma



- 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

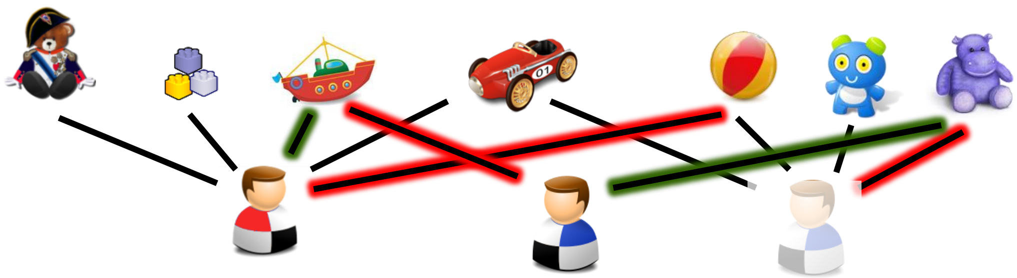
A Key Lemma

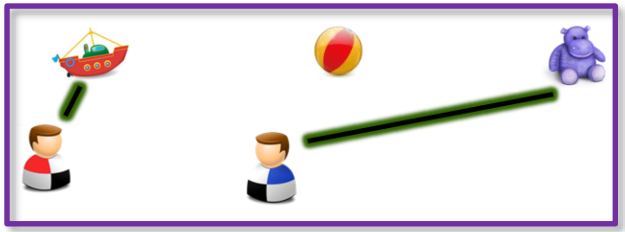




- Consider an optimal allocation (w.r.t. some declared types)
- Ignore the goods that are not allocated,
  - and hence that cannot be verified later…
- Focus on an arbitrary coalition of agents
- In this novel setting, compute an optimal allocation

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**

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 $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. | $\lfloor$ Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
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. | 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

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 $\langle \mathcal{C}, \texttt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
For each agent $i \in \mathcal{A}$,
  For each set $\mathcal{C} \in \mathbb{C}$,
    Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
    Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
  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 $p_i^{\xi}(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

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. $\lfloor$ 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

«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 $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

---

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 $\langle \mathcal{C}, \texttt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. For each set $\mathcal{C} \in \mathbb{C}$,
6. Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
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)

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 $\mathbf{v}(\pi) = (v_1, \ldots, v_n)$;

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 $\langle \mathcal{C}, \texttt{img}(\pi), \omega \rangle$ w.r.t. $\mathbf{w}$;
4. For each agent $i \in \mathcal{A}$,
5. | For each set $\mathcal{C} \in \mathbb{C}$,
6. | | Let $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C}}, (v_i, \mathbf{w}_{-i}))$; $(= v_i(\pi_{\mathcal{C}}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C}}))$;
7. | | Let $\Delta^2_{\mathcal{C},i}(\pi, \mathbf{w}) := \texttt{val}(\pi_{\mathcal{C} \setminus \{i\}}, \mathbf{w})$; $(= \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_{\mathcal{C} \setminus \{i\}}))$;
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 $\mathbf{v}(\pi) = (v_1, \ldots, v_n)$;

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 $\Delta^1_{\mathcal{C},i}(\pi, \mathbf{w}) := \mathtt{val}(\pi_\mathcal{C}, (v_i, \mathbf{w}_{-i}))$; $\quad (= v_i(\pi_\mathcal{C}) + \sum_{j \in \mathcal{C} \setminus \{i\}} w_j(\pi_\mathcal{C}))$;
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

Does not depend on *i*

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

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 $\mathbf{v}(\pi) = (v_1, ..., v_n)$;

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 $p_i^\xi(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

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

Relevant Properties of the Shapley Value

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

(II) If φ is *supermodular* (resp., *submodular*), then $\sum_{i \in R} \phi_i(\mathcal{G}) \geq \varphi(R)$ (resp., $\sum_{i \in R} \phi_i(\mathcal{G}) \leq \varphi(R)$), for each coalition $R \subseteq N$.

(III) If $\mathcal{G}' = \langle N, \varphi' \rangle$ is a game such that $\varphi'(R) \geq \varphi(R)$, for each $R \subseteq N$, then $\phi_i(\mathcal{G}') \geq \phi_i(\mathcal{G})$, for each agent $i \in N$.

Core Allocation

$$\varphi(R \cup T) + \varphi(R \cap T) \geq \varphi(R) + \varphi(T) \;\; \text{(resp., } \varphi(R \cup T) + \varphi(R \cap T) \leq \varphi(R) + \varphi(T)\text{)}$$

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.** $\left\{ \begin{array}{l} \textbf{selected products} \\ \quad \textit{and} \\ \textbf{verified values} \end{array} \right.$

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

Properties

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

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

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

Properties

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

The game is supermodular;
so the Shapley value is stable

Further Observations for Fairness

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

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

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

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

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

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

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

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

Further Observations for Fairness

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

$$\pi \geq \pi'$$

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

Further Observations for Fairness

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

$$\pi \geq \pi'$$

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

Efficiency  **Fairness**

Outline

- Background on Mechanism Design
- Mechanisms for Allocation Problems
- Complexity Analysis
- Case Study

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
- However…

Probabilistic Computation

- #P-complete
- However…

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]

Probabilistic Computation

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 $p^{\xi}_i(\pi, \mathbf{w}) := \xi_i(\pi, \mathbf{w}) - v_i(\pi)$;

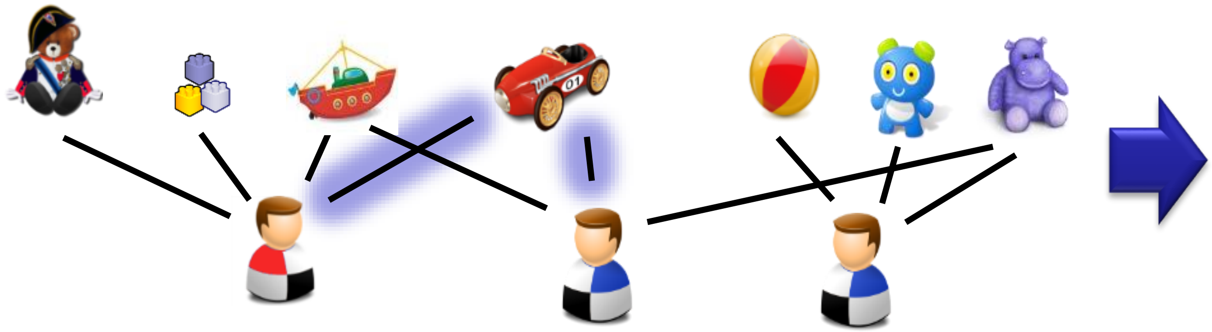
Use sampling, rather than exaustive search.

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

Back to Exact Computation: Islands of Tractability

- Can we find classes of instances for «allocation games» over which the Shapley value can be efficiently computed?

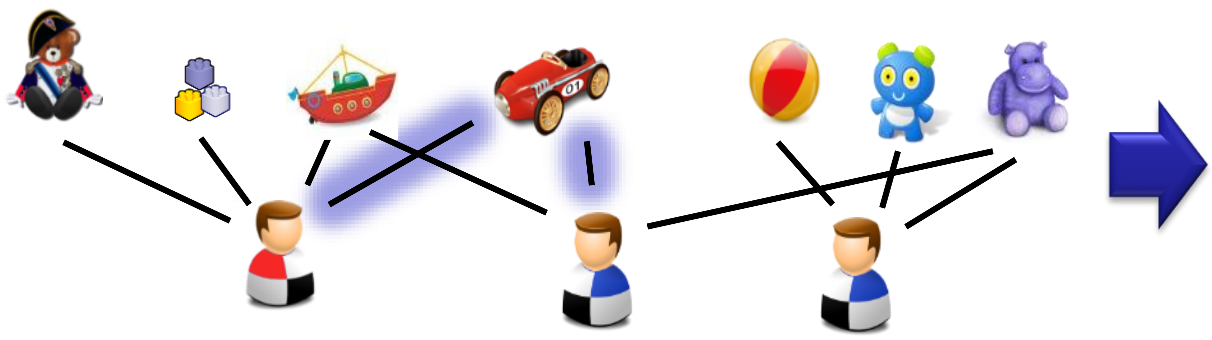
Bounded Sharing Degree



Sharing degree = 2

- Sharing degree
  - Maximum number of agents competing for the same good

Bounded Sharing Degree



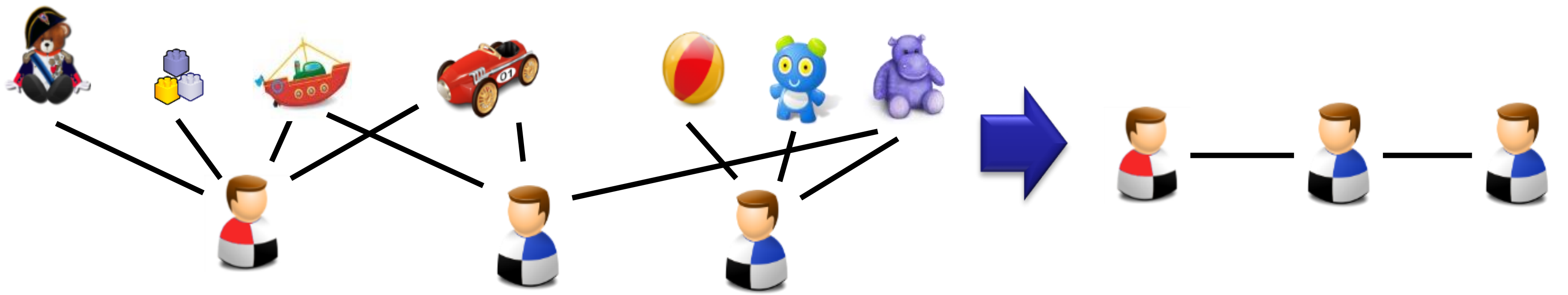
Sharing degree = 2

- Sharing degree
  - Maximum number of agents competing for the same good

The Shapley value can be computed in polynomial time whenever the sharing degree is 2 at most.

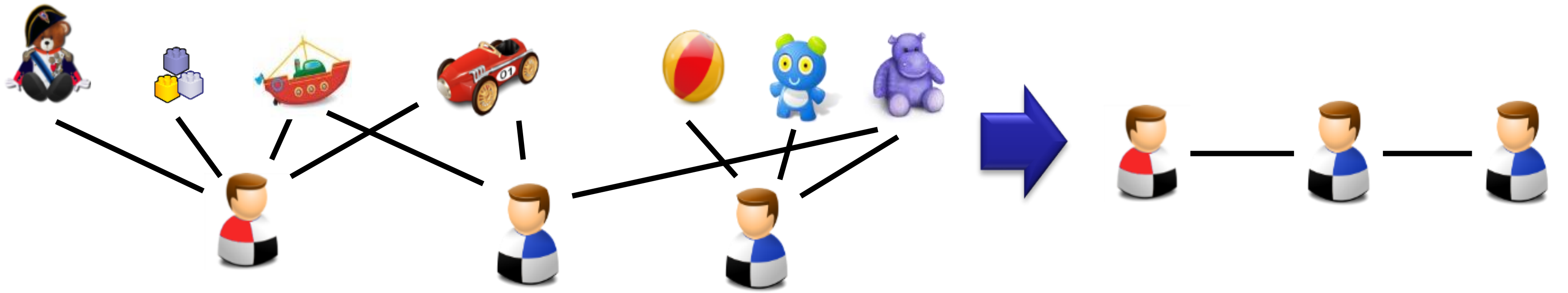
Bounded Interactions

Bounded Interactions



- Interaction graph
  - There is an edge between any pair of agents competing for the same good

Bounded Interactions



- Interaction graph
  - There is an edge between any pair of agents competing for the same good

The Shapley value can be computed in polynomial time whenever the interaction graph is a tree.

or, more generally, if it has bounded treewidth

Outline

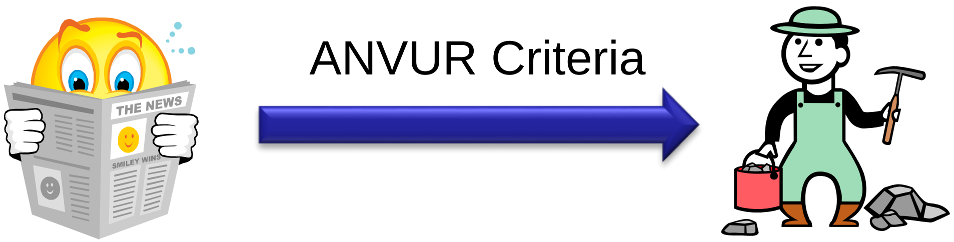
- Background on Mechanism Design
- Mechanisms for Allocation Problems
- Complexity Analysis
- **Case Study**

Case Study: Italian Research Assessment Program

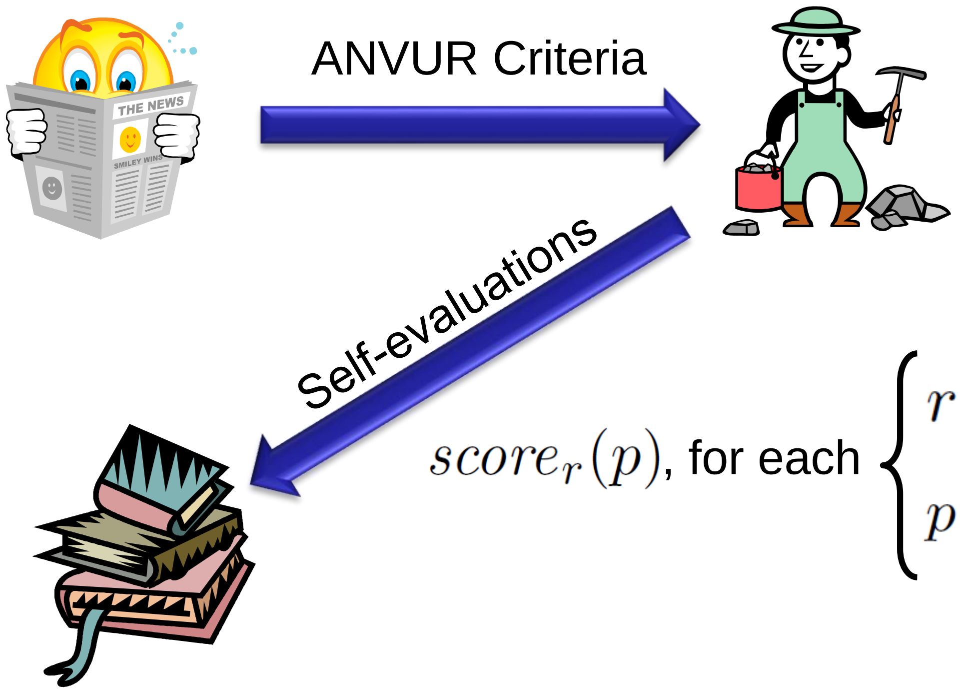
- VQR: ANVUR should evaluate the quality of research of all Italian research structures
- Funds for the structures in the next years depend on the outcome of this evaluation
- Substructures will be also evaluated (departments)

(1) 2004-2010
(2) 2011-2014

ANVUR Evaluation

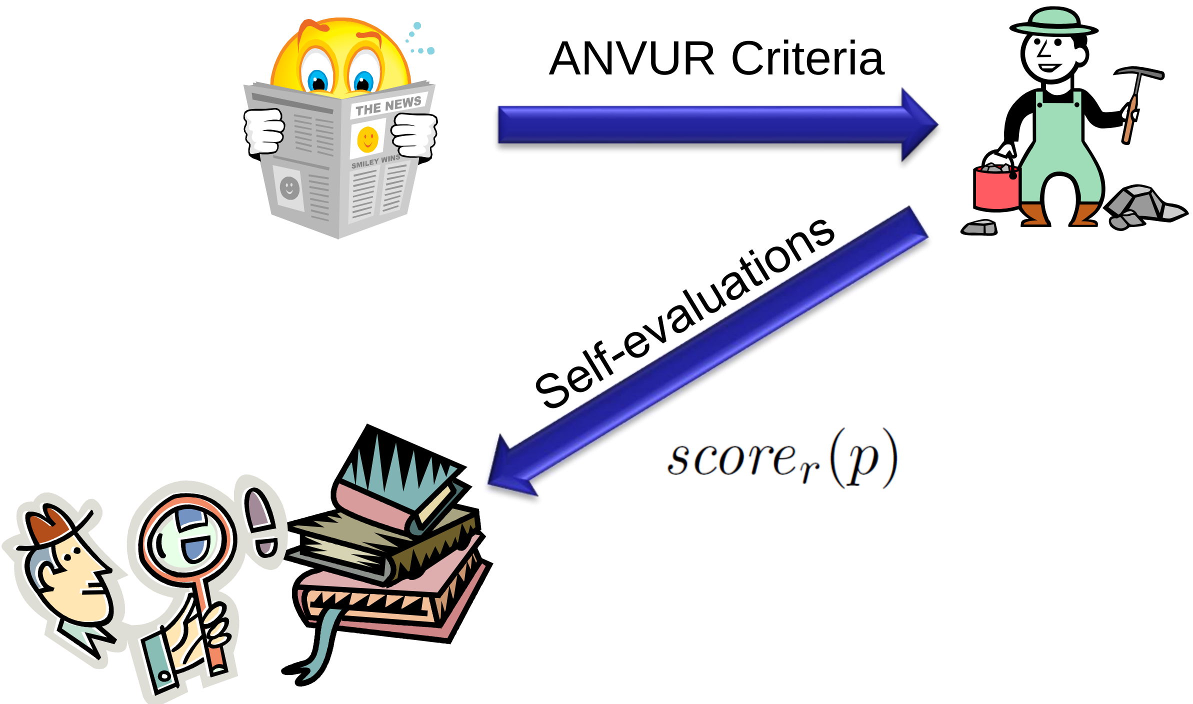


ANVUR Evaluation



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

ANVUR Evaluation

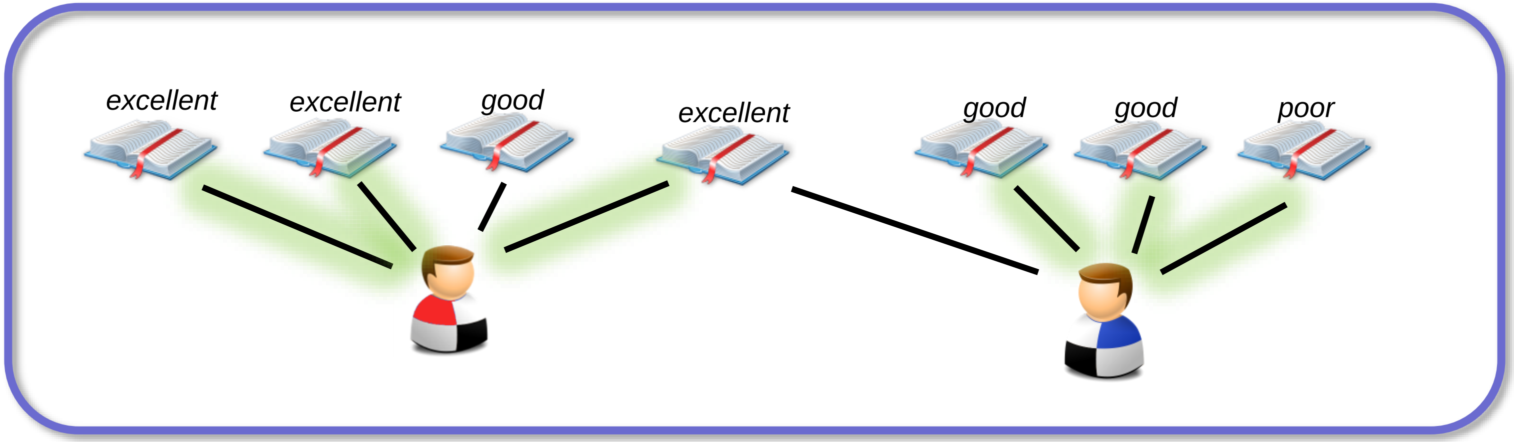


Structures are in charge of selecting the products to submit

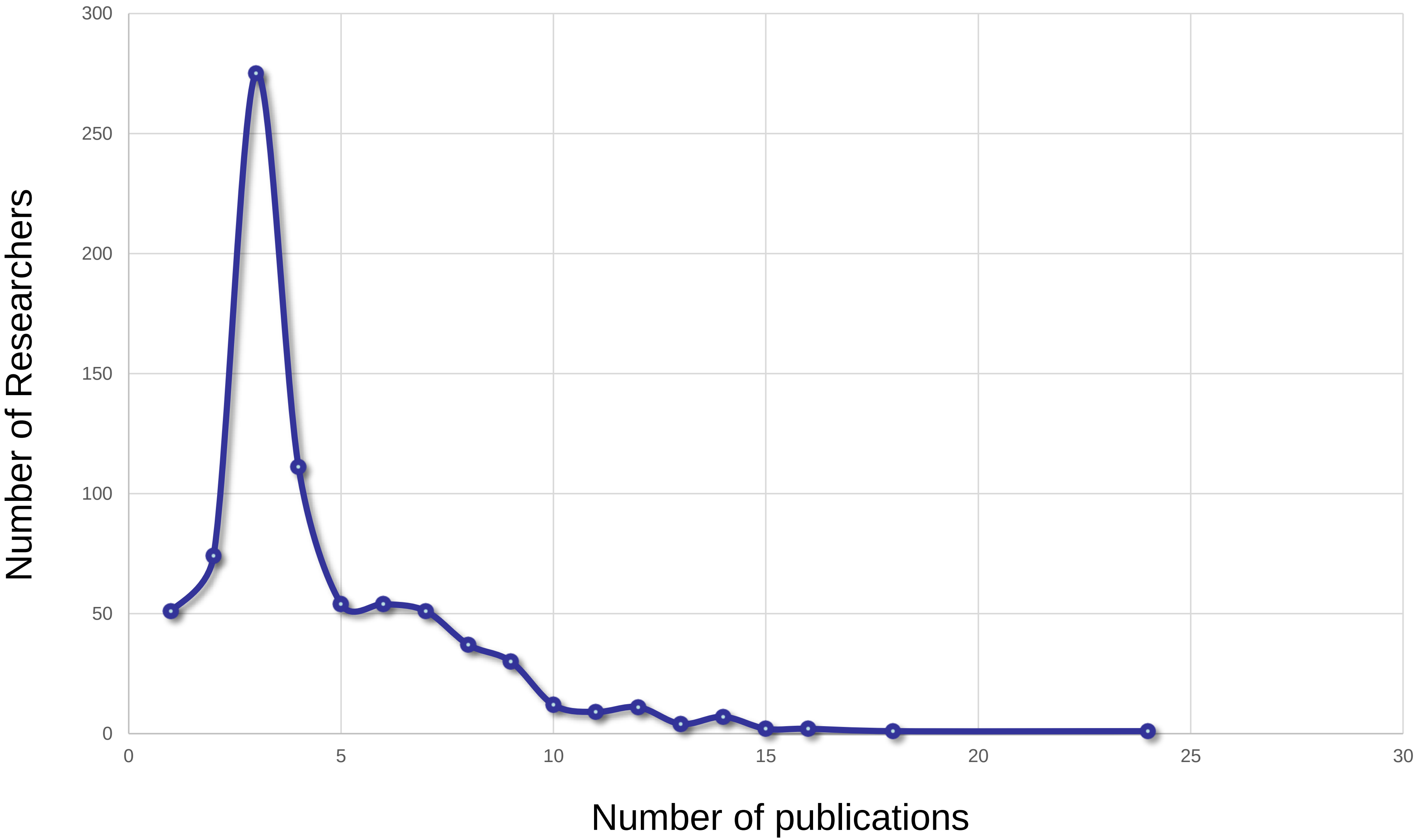
Constraints (2004-2010)

- Every researcher has to submit 3 publications
- A publication cannot be allocated to two researchers

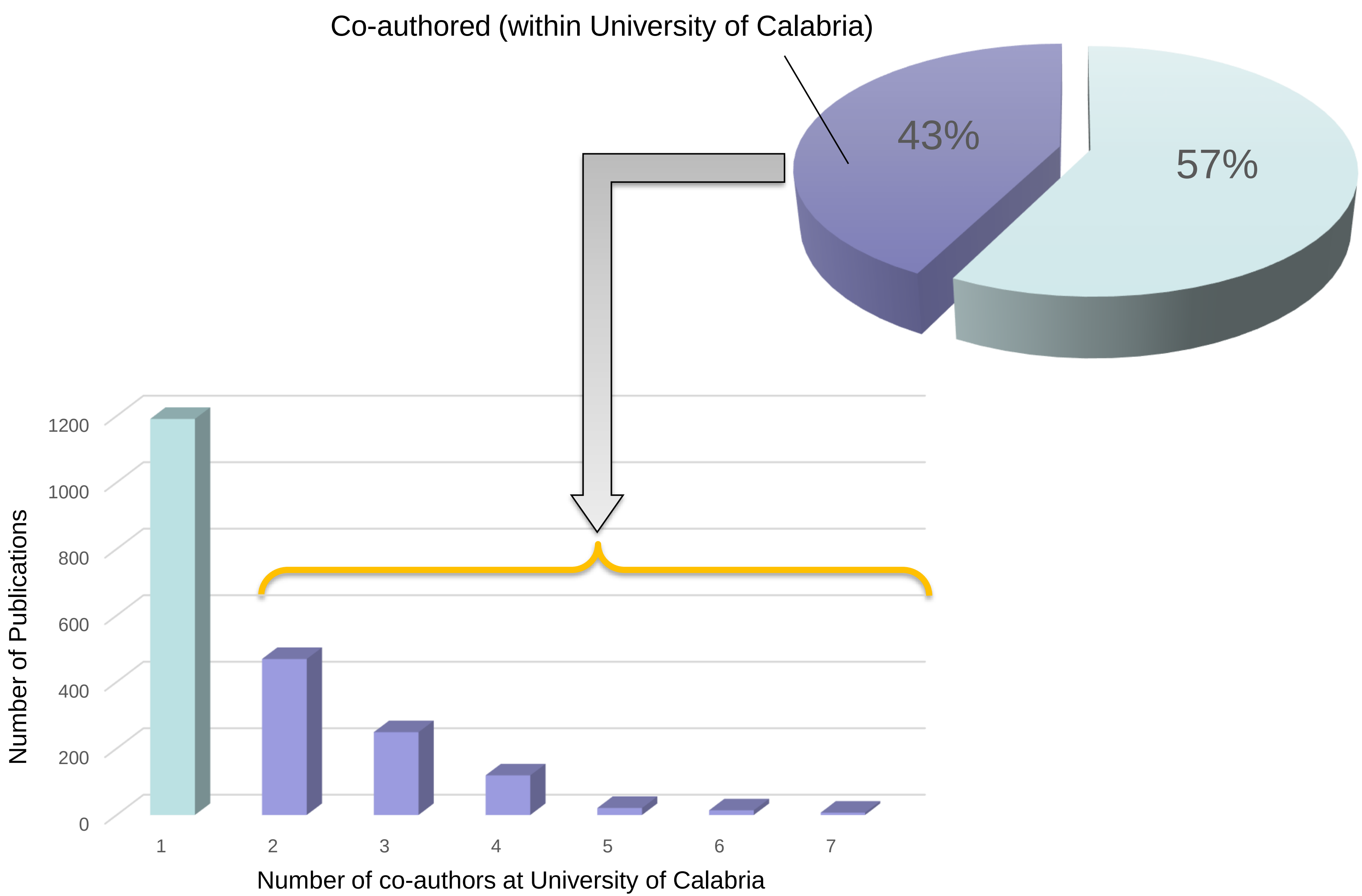
Allocation Problem



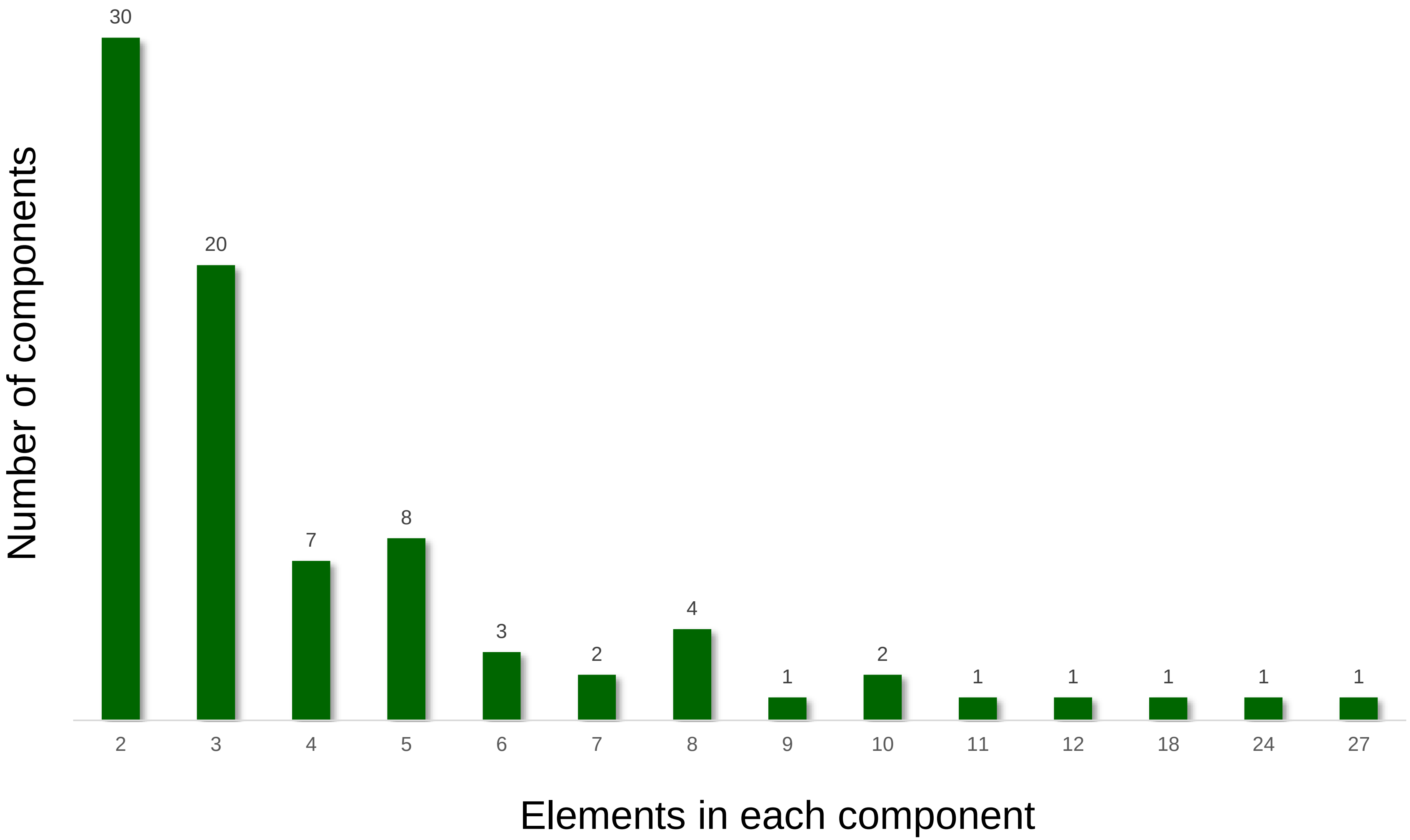
Co-Autorships at University of Calabria



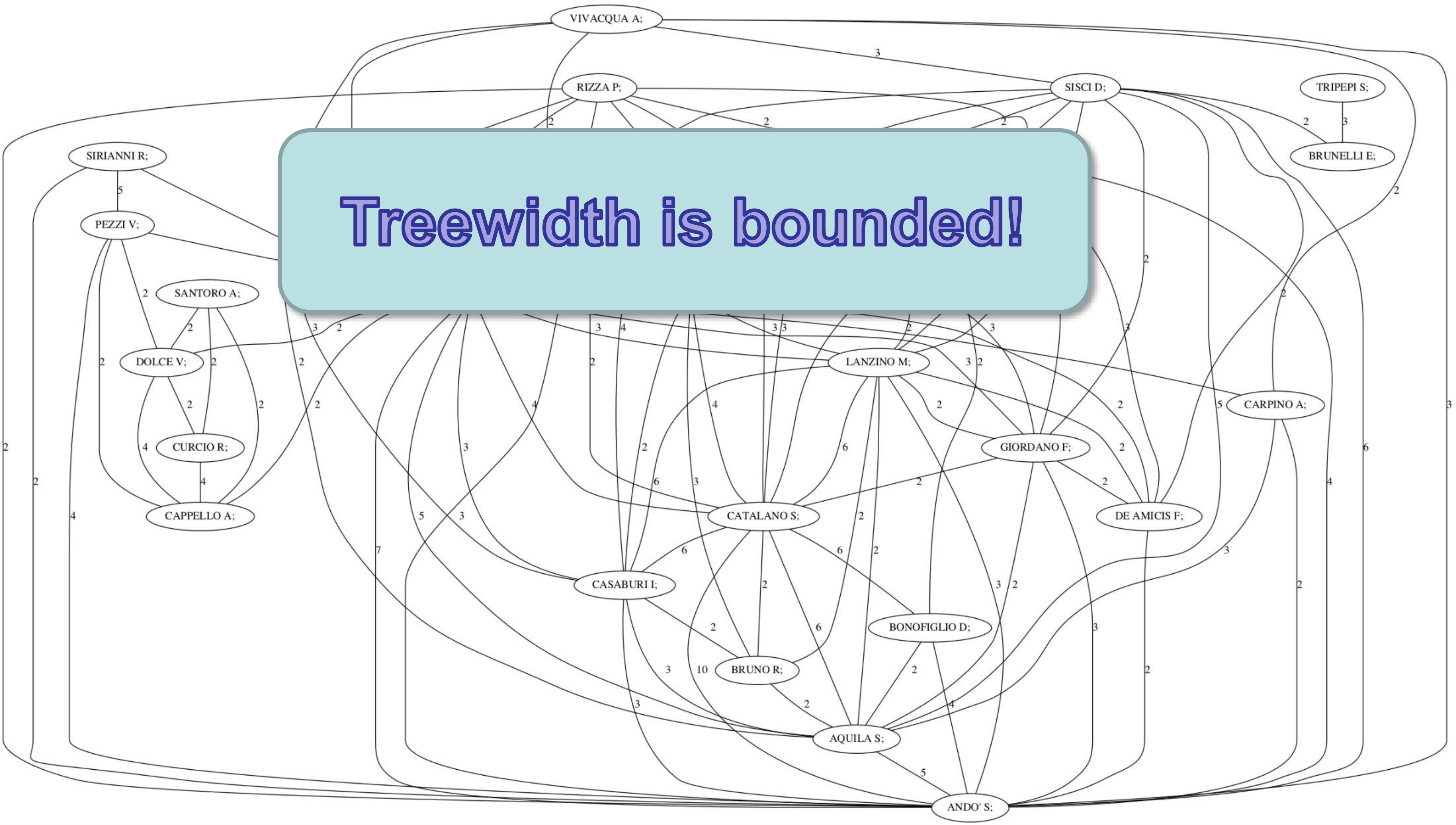
Co-Autorships at University of Calabria



Components at University of Calabria

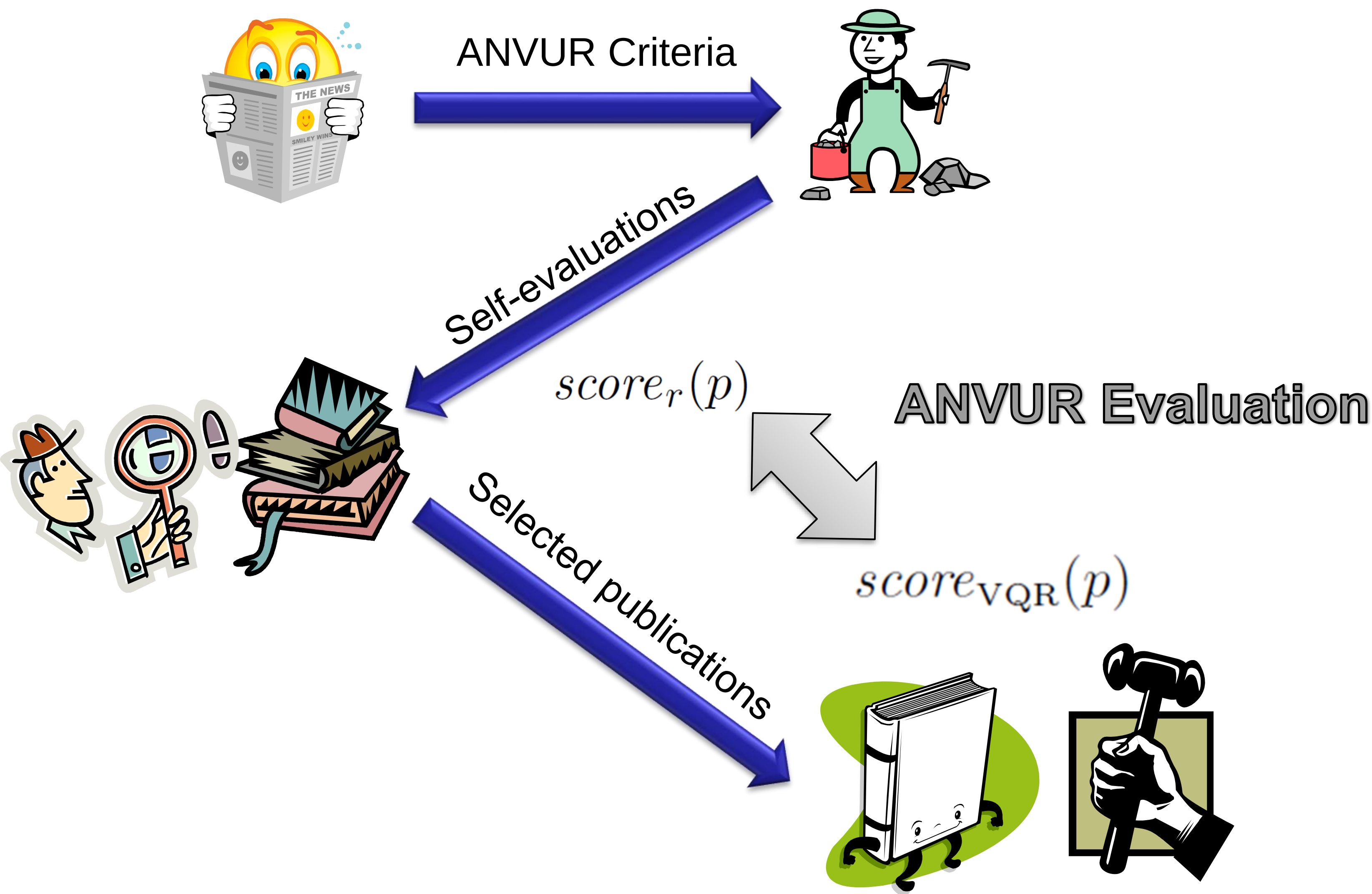


An Example Component

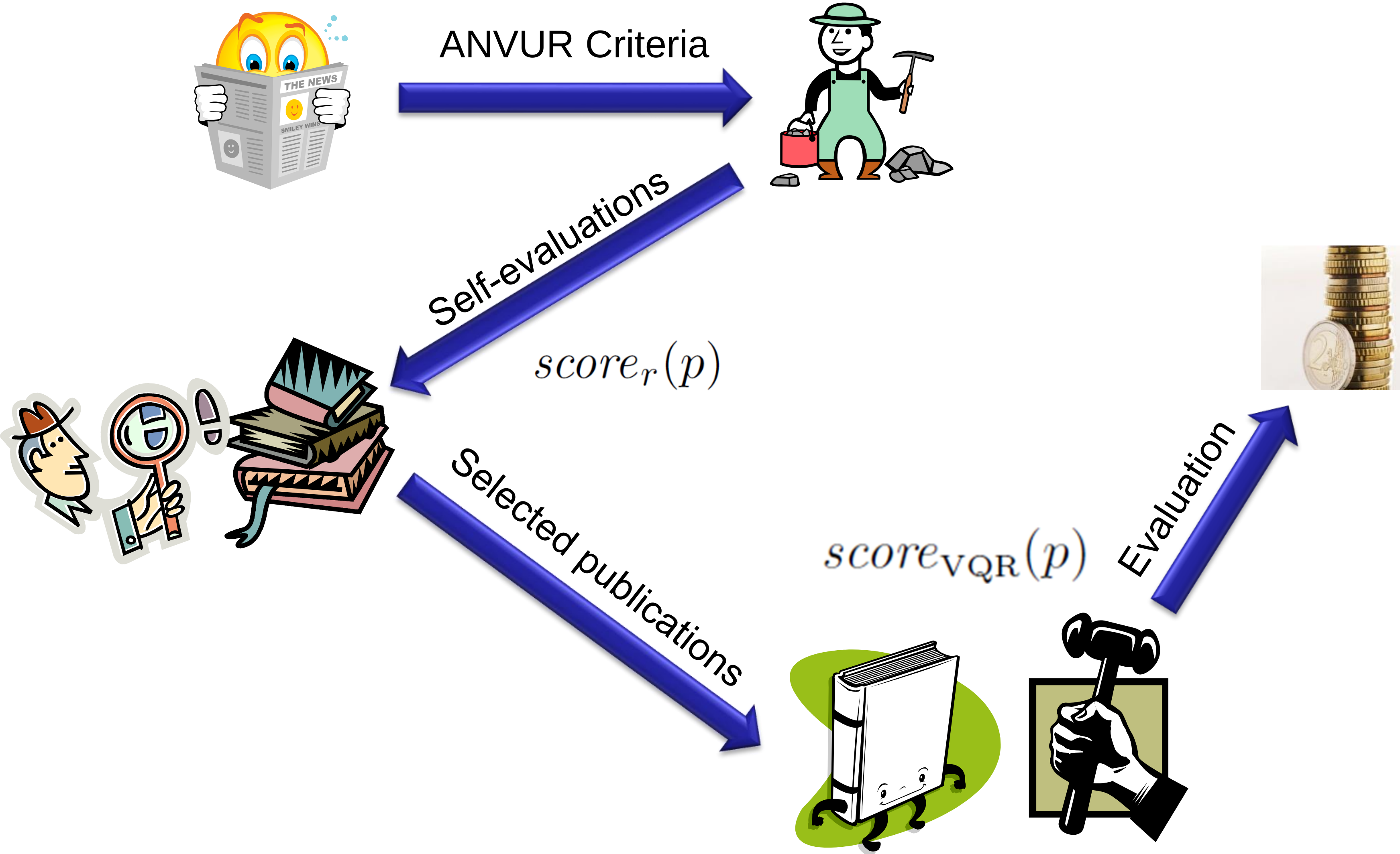


Treewidth is bounded!

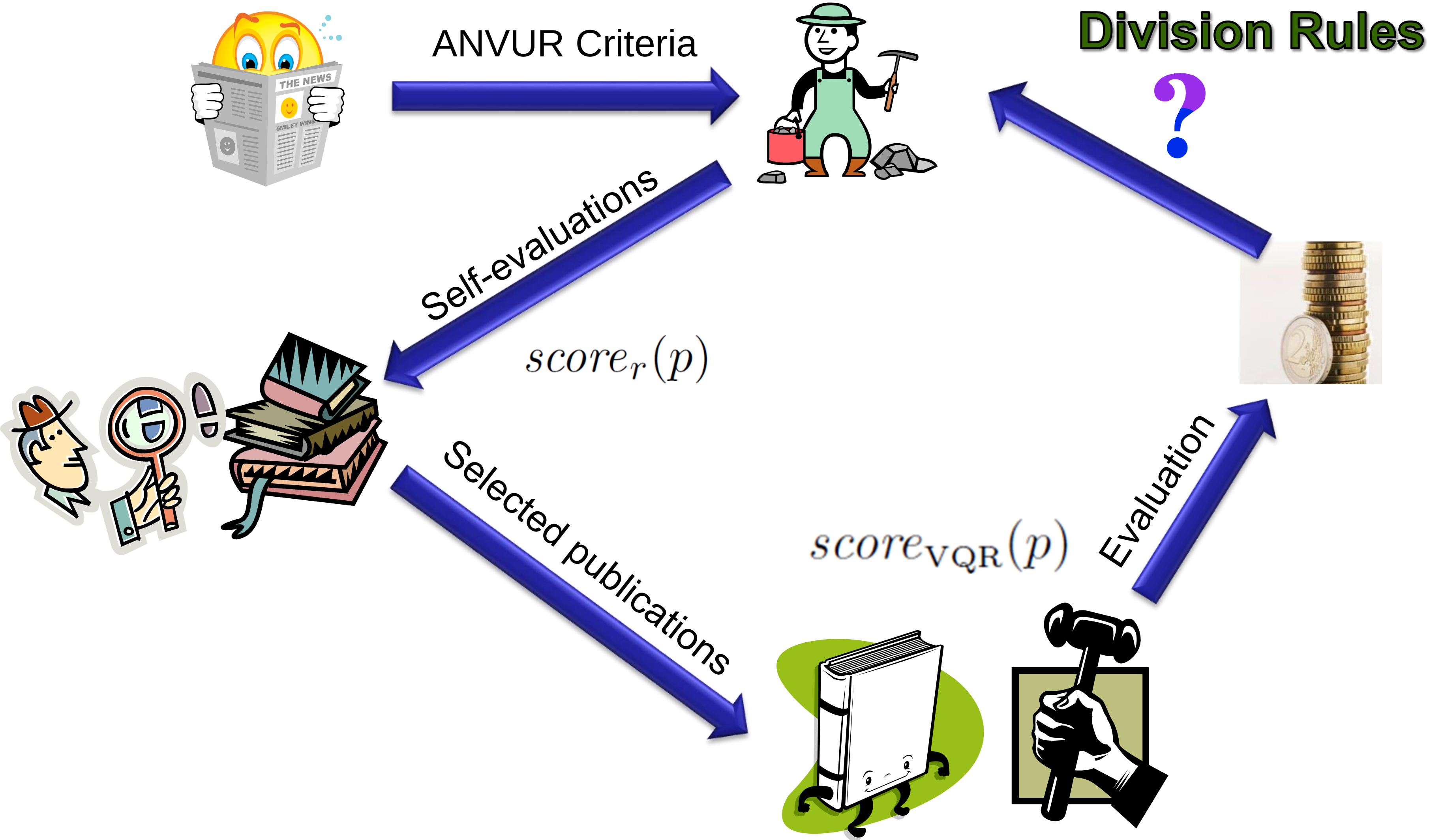
ANVUR Evaluation



ANVUR Evaluation



ANVUR Evaluation



Issues

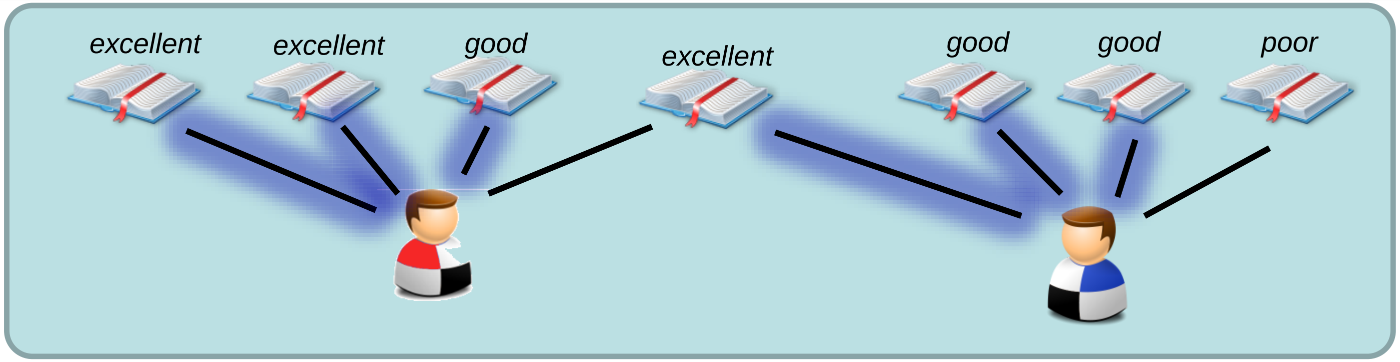
- Allocation Problem
- Valuations are declared (punishments?)
- The program is meant to evaluate the structures…
  - …but outcomes are used to evaluate researchers, too

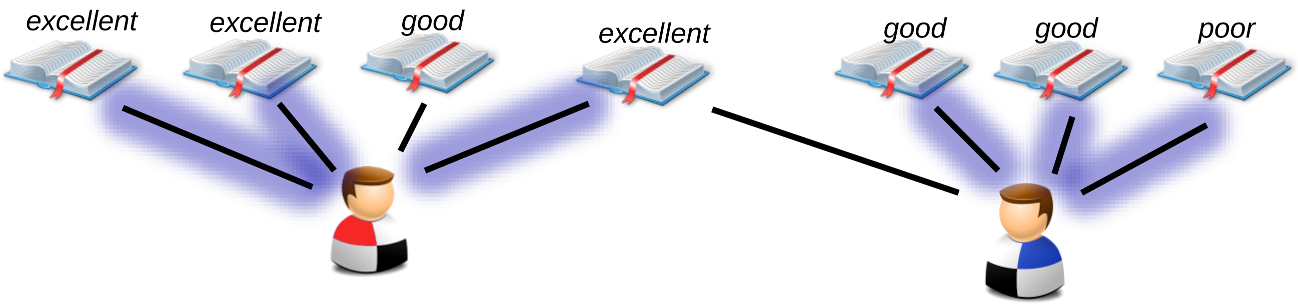
Global Evaluation

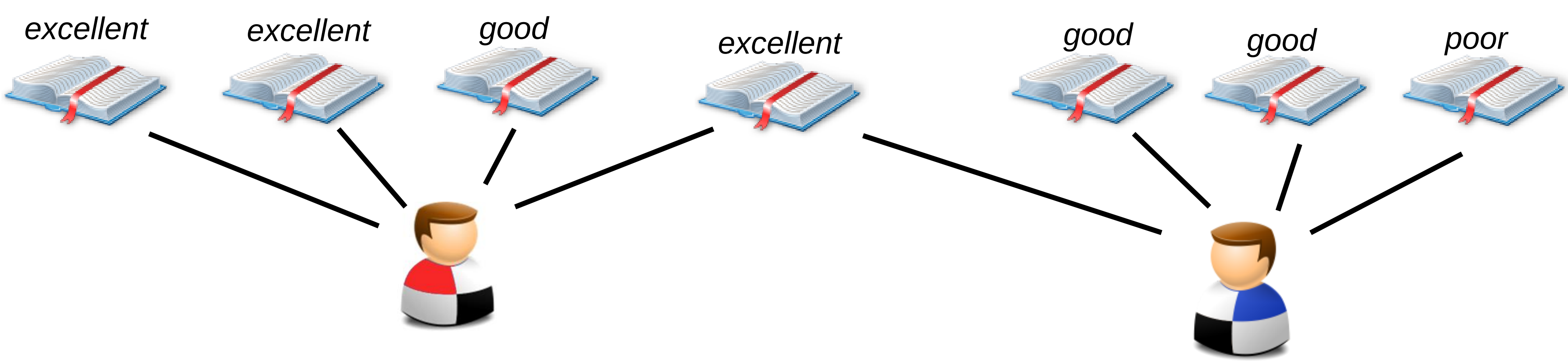
Individual Evaluations

A Closer Look

Optimal Allocation

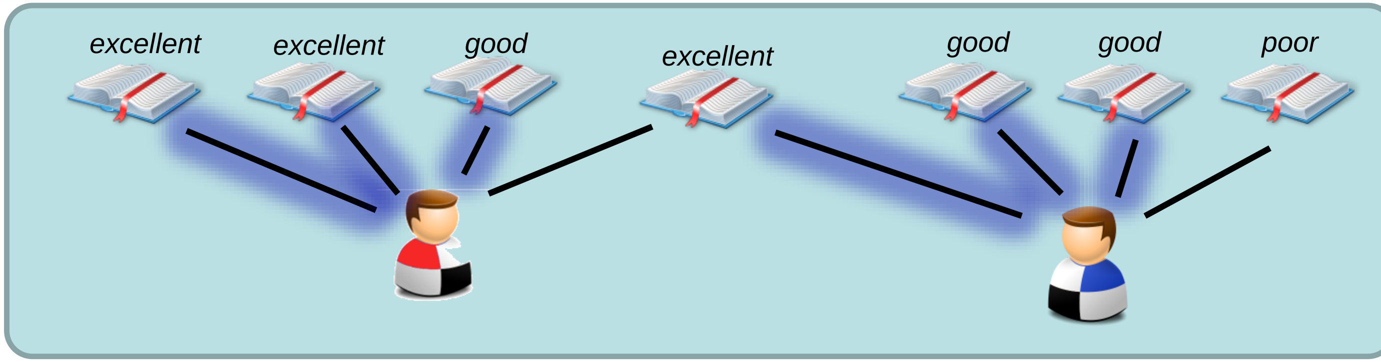


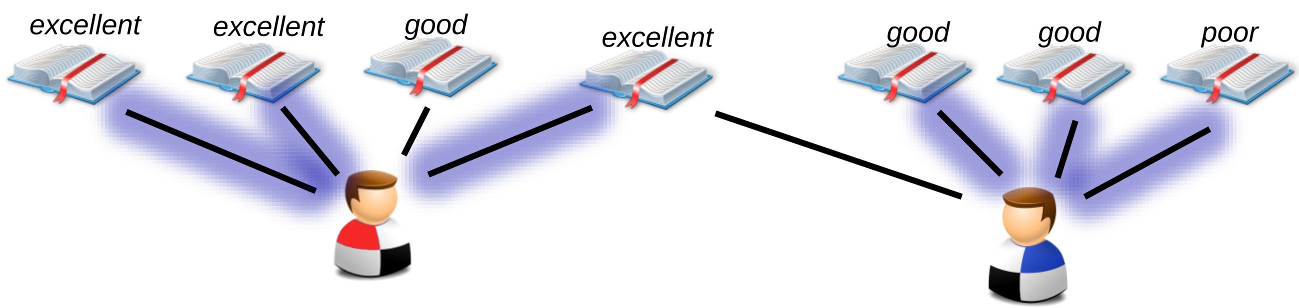




A Closer Look

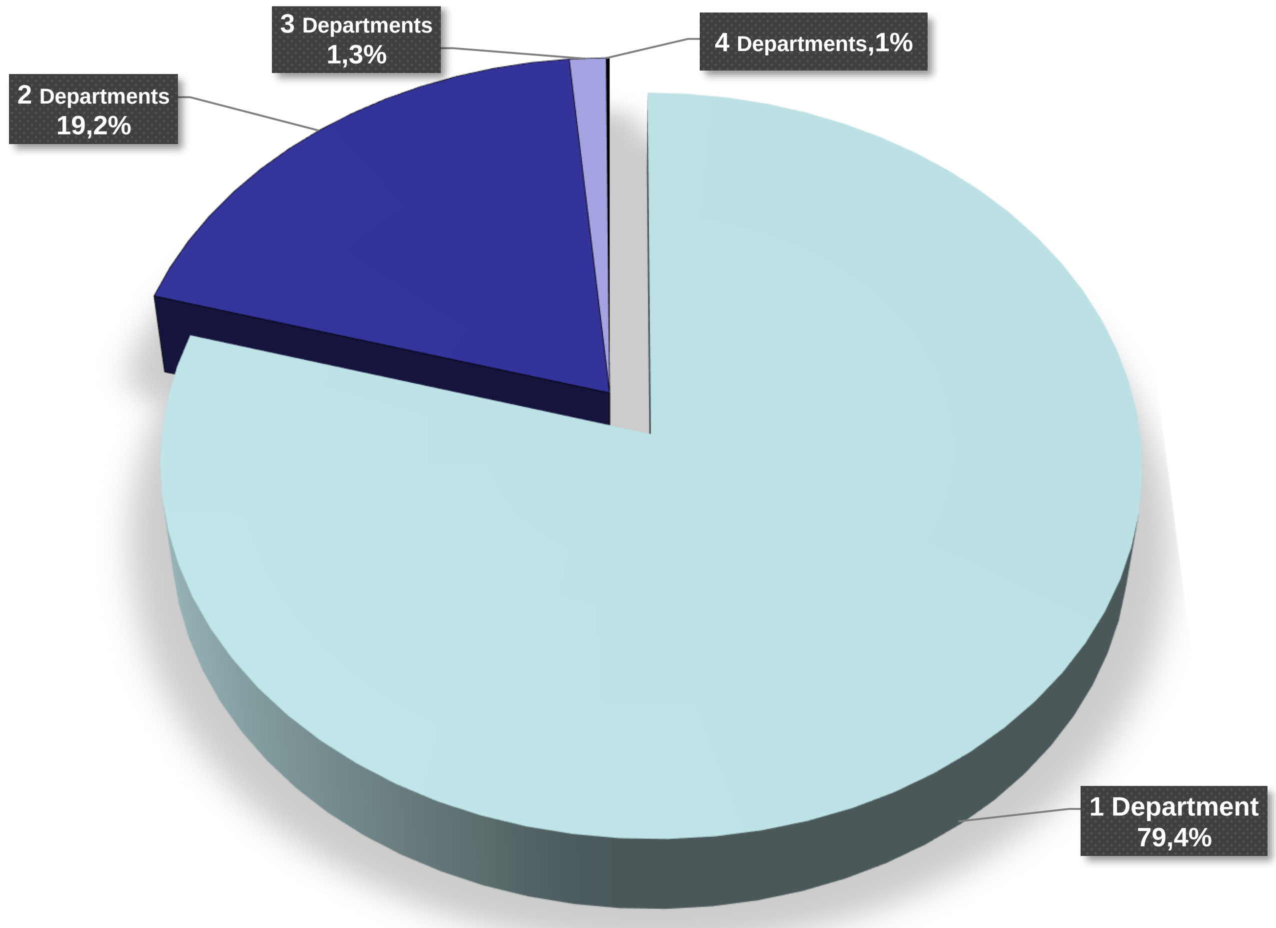
Optimal Allocation





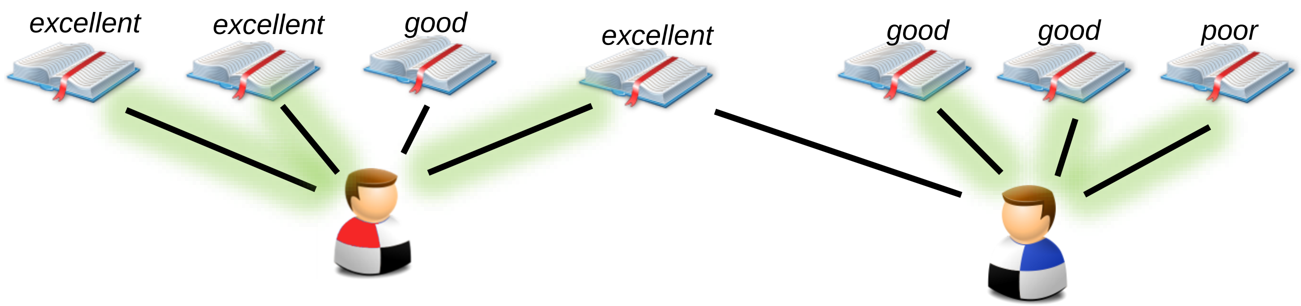
- «Penalizing»  is not fair!
- Unless it is clear that no penalization will occur,  will act «strategically»

Distribution at University of Calabria



The Story….

- 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, «agreements» have been made



- Conflicts resolved «strategically», «hierarchically», …

The optimum has been missed!

No fairness at all!

Side Results

- University of Rome uses (parts of) our findings
- University of Calabria uses (parts of) our findings
- Head of the «Presidio della Qualità» at University of Calabria
- Still trying to generalize at national level….

Thank you!