Fairness in allocation problems

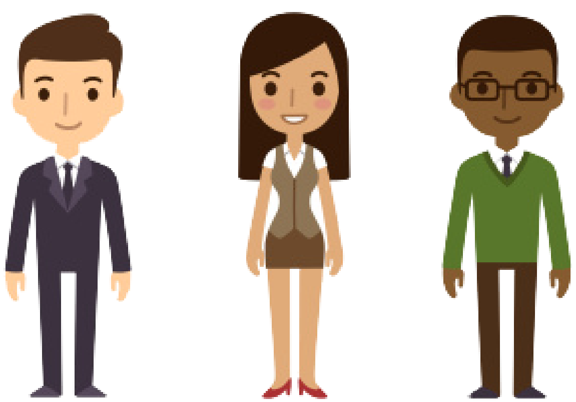
Ioannis Caragiannis
University of Patras

Seminar at GSSI, L'Aquila, 2-4 July, 2018

Allocations of goods

- **Indivisible** goods

- **Agents** with additive **valuations** for goods



- Goal: **divide** the goods **fairly**

Formally …

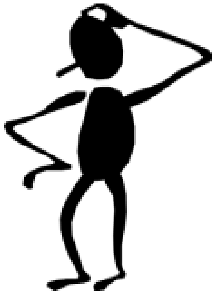
- n **agents**
- A set of **goods** G
- Agent i has **valuation** $u_i(g)$ for good g
- Valuations are additive, i.e.,

$$v_i(S) = \sum_{g \in S} v_i(g)$$

- **Allocation**: a partition $A=(A_1, \ldots, A_n)$ of the goods in G

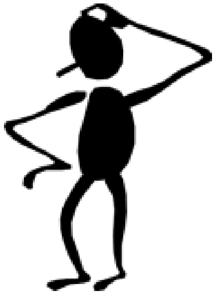
Maxmin fair allocations

What does “**fairly**” mean?



- **Fairness notions**
  - Envy freeness
  - Proportionality
  - **Maxmin share (MmS) allocation**

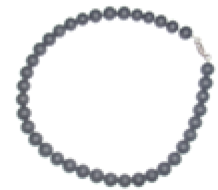
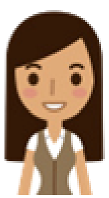
What does “fairly” mean?



- **Fairness notions**
  - **Envy freeness**: every agent prefers her own bundle to the bundle of any other agent
  - **Proportionality**: every agent feels that she gets at least 1/n-th of the goods
  - **Maxmin share (MmS) allocation**: each agent’s value is at least the best guarantee when dividing the goods into n bundles and getting the least valuable bundle

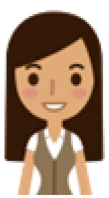
$$\forall i, v_i(A_i) \geq \theta_i = \max_{A'} \min_{j \in N} v_i\left(A'_j\right)$$

MmS: an example

			ROLLS ROYCE		
	$500	$600	$200	$400	$300
	$700	$700	$300	$200	$100
	$900	$600	$200	$200	$100

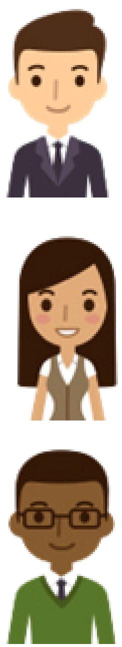
MmS: an example

Let’s compute the MmS thresholds first

						θ_i
	\$500	\$600	\$200	\$400	\$300	
	\$700	\$700	\$300	\$200	\$100	
	\$900	\$600	\$200	\$200	\$100	

MmS: an example

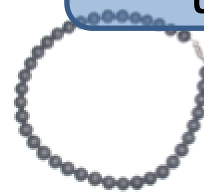
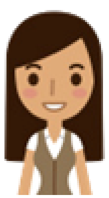
Let's compute the MmS thresholds first



					θ_i
\$500	\$600	\$200	\$400	\$300	\$600
\$700	\$700	\$300	\$200	\$100	\$600
\$900	\$600	\$200	\$200	\$100	\$500

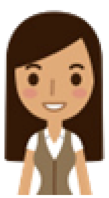
MmS: an example

Now, let's compute the allocation

						θ_i
	\$500	\$600	\$200	\$400	\$300	\$600
	\$700	\$700	\$300	\$200	\$100	\$600
	\$900	\$600	\$200	\$200	\$100	\$500

MmS: an example

Now, let's compute the allocation

						θ_i
	\$500	**\$600**	\$200	\$400	\$300	\$600
	**\$700**	\$700	\$300	\$200	\$100	\$600
	\$900	\$600	**\$200**	**\$200**	**\$100**	\$500

An implication

- **Theorem**: Proportionality implies MmS

An implication

- **Theorem**: Proportionality implies MmS
- **Proof**: Let A be a proportional allocation. Then,

$$\forall i, v_i(A_i) \geq \frac{1}{n} v_i(G)$$

But the MmS threshold for agent i is

$$\theta_i = \max_{A'} \min_{j \in N} v_i(A'_j) \leq \frac{1}{n} v_i(G)$$

Hence,

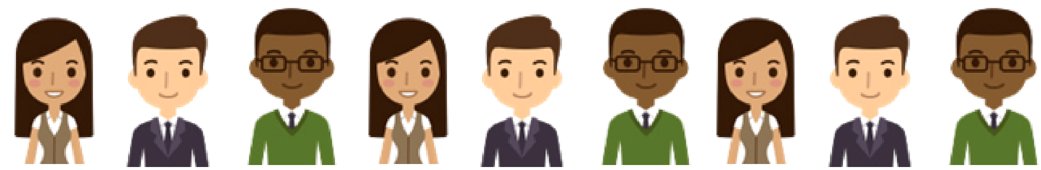
$$\boxed{\forall i, v_i(A_i) \geq \theta_i}$$

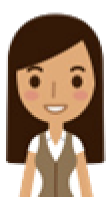
Related work

- MmS concept introduced by Budish (2011)
- **NP-hard** via a reduction from Partition
- MmS allocations **may not exist** but there is always a 2/3-**approximation**
  - Procaccia & Wang (2014)
  - Kurokawa, Procaccia, & Wang (2018)
- Polynomial-time 2/3-approximation algorithms
  - Amanatidis, Markakis, Nikzad, and Saberi (2017)
  - Barman & Murthy (2017)
- Best possible bound: 3/4-approximation algorithm
  - Ghodsi, Hajiaghayi, Seddighin, Seddighin, & Yami (2018)

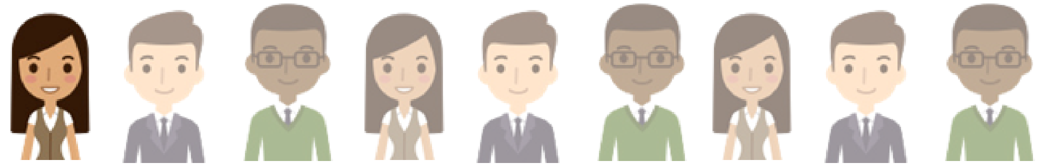
A 1/2-approximate MmS algorithm

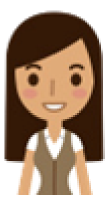
The draft mechanism

- Drafting order: 

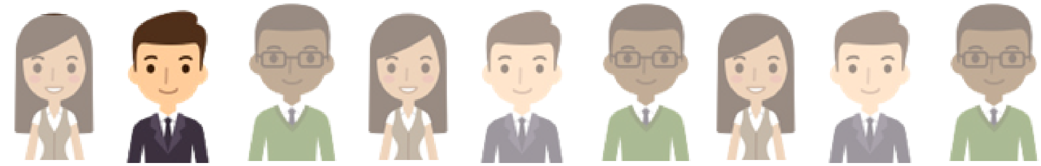
	\$1200	\$200	\$300	\$200	\$100
	\$800	\$500	\$200	\$300	\$200
	\$800	\$400	\$400	\$200	\$200

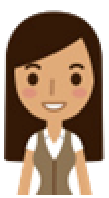
The draft mechanism

- Drafting order: 

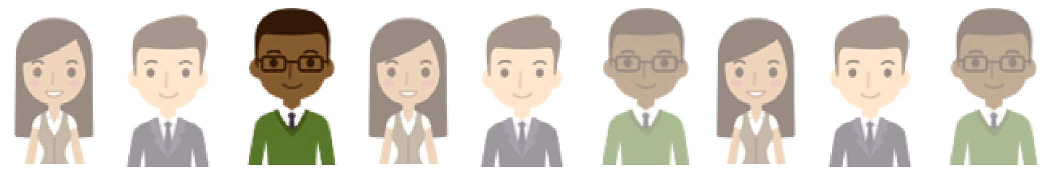
	$1200	$200	$300	$200	$100
	**$800**	$500	$200	$300	$200
	$800	$400	$400	$200	$200

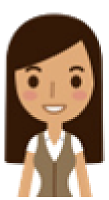
The draft mechanism

- Drafting order: 

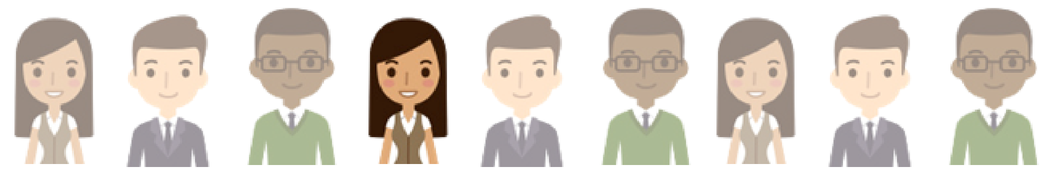
	$1200	$200	**$300**	$200	$100
	**$800**	$500	$200	$300	$200
	$800	$400	$400	$200	$200

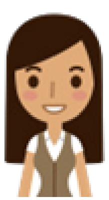
The draft mechanism

- Drafting order: 

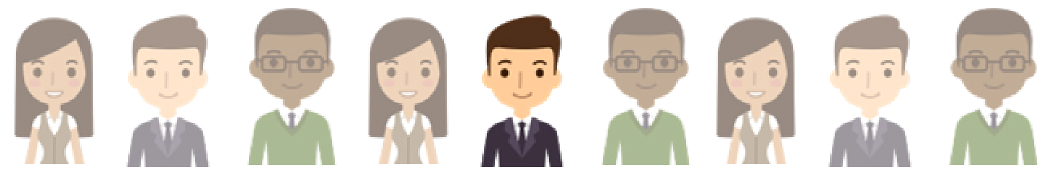
	$1200	$200	**$300**	$200	$100
	**$800**	$500	$200	$300	$200
	$800	**$400**	$400	$200	$200

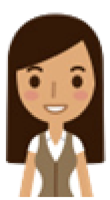
The draft mechanism

- Drafting order: 

	$1200	$200	**$300**	$200	$100
	**$800**	$500	$200	**$300**	$200
	$800	**$400**	$400	$200	$200

The draft mechanism

- Drafting order: 

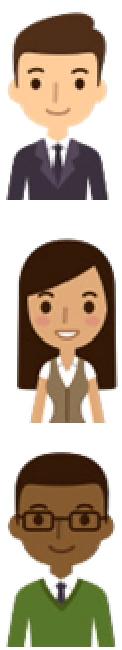
	$1200	$200	**$300**	$200	**$100**
	**$800**	$500	$200	**$300**	$200
	$800	**$400**	$400	$200	$200

The draft mechanism

- Drafting order: 

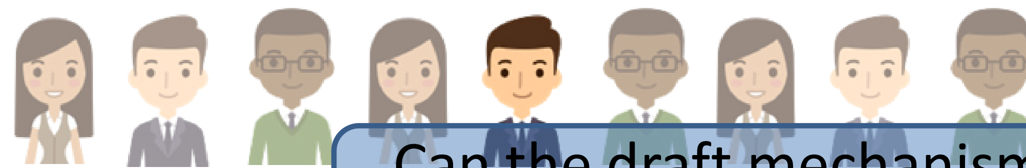
Can the draft mechanism compute MmS allocations?

θ_i

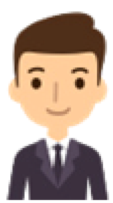
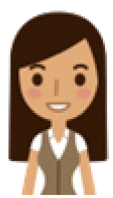


$1200	$200	**$300**	$200	**$100**
$800	$500	$200	**$300**	$200
$800	**$400**	$400	$200	$200

The draft mechanism

- Drafting order: 

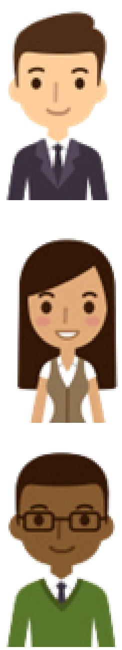
Can the draft mechanism compute MmS allocations?

						θ_i
	$1200	$200	**$300**	$200	**$100**	$400
	**$800**	$500	$200	**$300**	$200	$500
	$800	**$400**	$400	$200	$200	$600

The draft mechanism

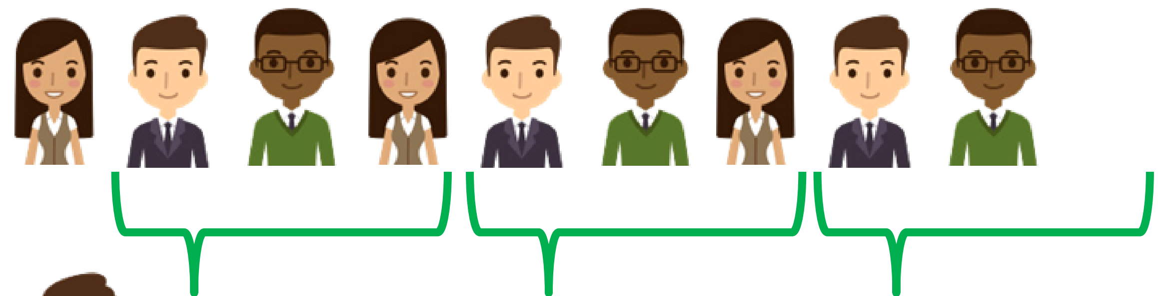
- Drafting order: 

Can the draft mechanism compute MmS allocations?



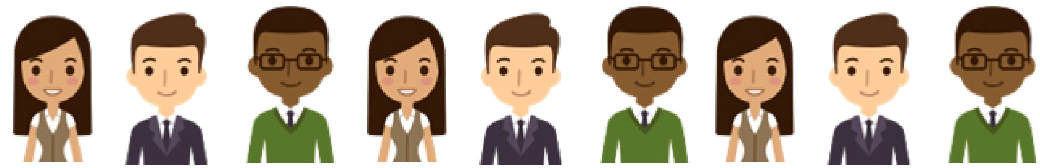
						θ_i
	$1200	$200	**$300**	$200	**$100**	$400
	$800	$500	$200	**$300**	$200	$500
	$800	**$400**	$400	$200	$200	$600

The draft mechanism

- Drafting order: 
- **Phases** for agent 
- In each phase,  prefers the good he gets to the good every other agent gets
- So,

$$v_i(A_i) \geq \frac{1}{n} v_i(G) - \max_{g \in G} v_i(g)$$

The draft mechanism

- Drafting order: 

$$v_i(A_i) \geq \frac{1}{n} v_i(G) - \max_{g \in G} v_i(g)$$

- So, if $\max v_i(g) \leq \theta_i/2$, then $v_i(A_i) \geq \theta_i/2$ and the draft mechanism yields a 1/2-approximate MmS allocation

A 1/2-approximate MmS algorithm

- If there is an agent i and an good g with $v_i(g) \geq v_i(G)/2n \geq \theta_i/2$, **allocate good g to agent i**
- **Remove agent i and good g** from the instance and **repeat** until $v_i(g) \leq v_i(G')/2n'$ for every good g in the remaining instance with n' agents and set of goods G'
- **Run the draft mechanism** in the remaining instance

A 1/2-approximate MmS algorithm

- Key idea in the analysis:
  - **The MmS threshold increases** as we remove agents and goods

EF1: a relaxed version of EF

PROVABLY FAIR SOLUTIONS.

Spliddit offers quick, free solutions to everyday fair division problems, using methods that provide indisputable fairness guarantees and build on decades of research in economics, mathematics, and computer science.

Share Rent

Split Fare

Assign Credit

spliddit

- Fairness hierarchy
  1. Envy-freeness
  2. Proportionality
  3. Maxmin share guarantee
- Previous spliddit protocol
  - Find best fairness criterion
  - Maximize **social welfare** (subject to that criterion)

spliddit

Spliddit Feedback - [redacted]

admin@spliddit.org Jan 7

to admin

Hi! Great app :) We're 4 brothers that need to divide an inheritance of 30+ furniture items. This will save us a fist fight ;) I played around with the demo app and it seems there are non-optimal results for at least two cases where everyone distributes the same amount of value onto the same goods. Try it with either 3 people distributing 1000 points to good A and 0 to the 5 remaining goods, OR try 3 people, 5 goods, with everyone placing 200 on every good. The first case gives 0 to one person, 1 to another and 5 to the third. The second case gives 3 to one person and 1 to each of the others. Why is that? All the best, [redacted]

spliddit

Spliddit Feedback - [redacted]

admin@spliddit.org Jan 7

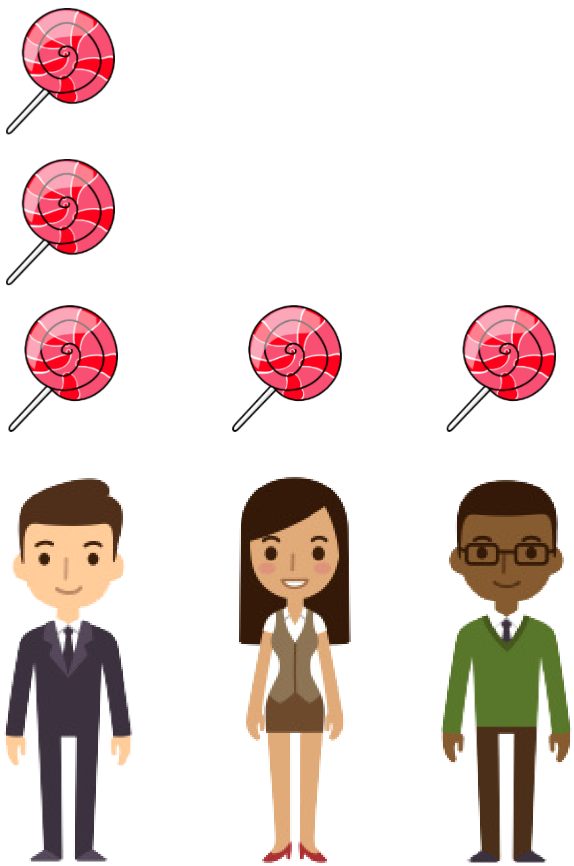
to admin

Hi! Great app :) We're 4 brothers that need to divide an inheritance of 30+ furniture items. This will save us a fist fight ;)

… *try 3 people, 5 goods, with everyone placing 200 on every good*.

… *gives 3 to one person and 1 to each of the others*. Why is that?

…



Relaxing EF

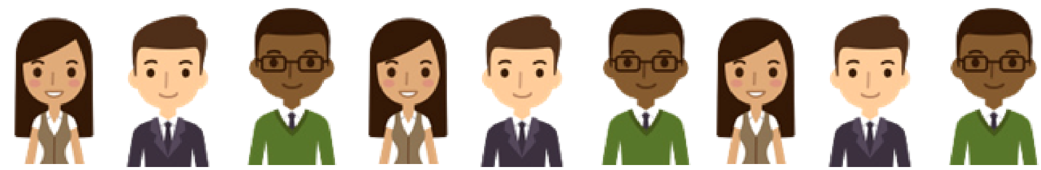
- **Envy-freeness up to one good (EF1)**:
  - There is a good that can be removed from the bundle of agent j so that any envy of agent i for agent j is eliminated

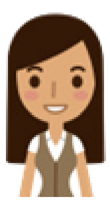
$$\forall i, j, \exists g \in A_j : v_i(A_i) \geq v_i(A_j - g)$$

Relaxing EF

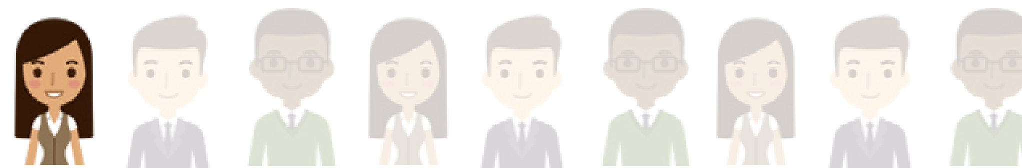
- **Envy-freeness up to one good (EF1)**:
  - There is a good that can be removed from the bundle of agent j so that agent i is not envious for agent j
  - Budish (2011)
  - Easy to achieve: **draft mechanism**
  - Also: Lipton, Markakis, Mossel, and Saberi (2004)

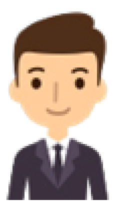
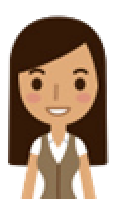
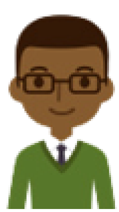
The draft mechanism

- Drafting order: 

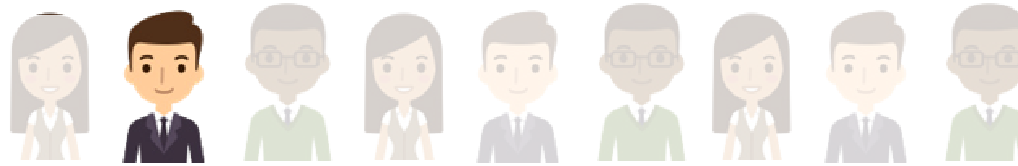
	\$1200	\$200	\$300	\$200	\$100
	\$800	\$500	\$200	\$300	\$200
	\$800	\$400	\$400	\$300	\$100

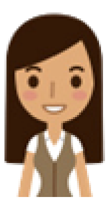
The draft mechanism

- Drafting order: 

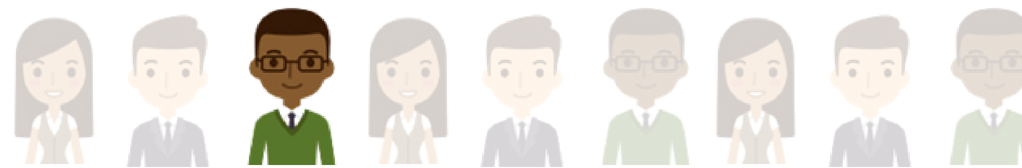
	$1200	$200	$300	$200	$100
	**$800**	$500	$200	$300	$200
	$800	$400	$400	$300	$100

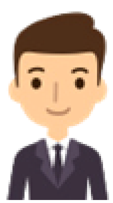
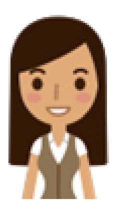
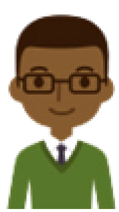
The draft mechanism

- Drafting order: 

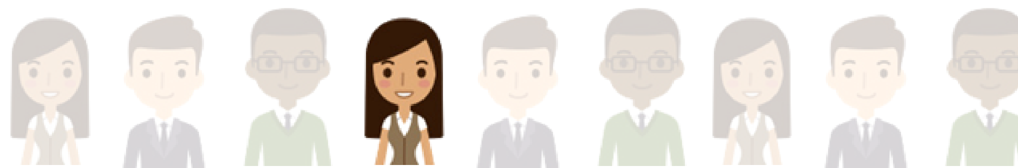
	$1200	$200	**$300**	$200	$100
	**$800**	$500	$200	$300	$200
	$800	$400	$400	$300	$100

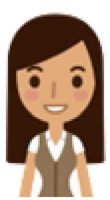
The draft mechanism

- Drafting order: 

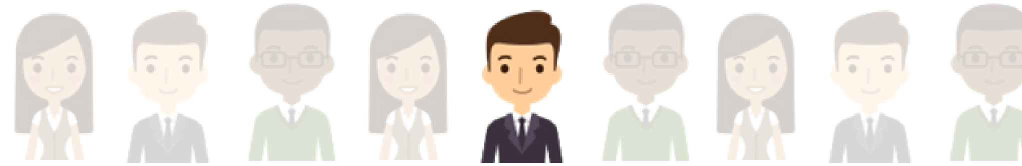
	$1200	$200	**$300**	$200	$100
	**$800**	$500	$200	$300	$200
	$800	**$400**	$400	$300	$100

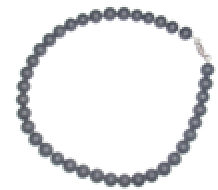
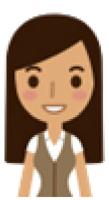
The draft mechanism

- Drafting order: 

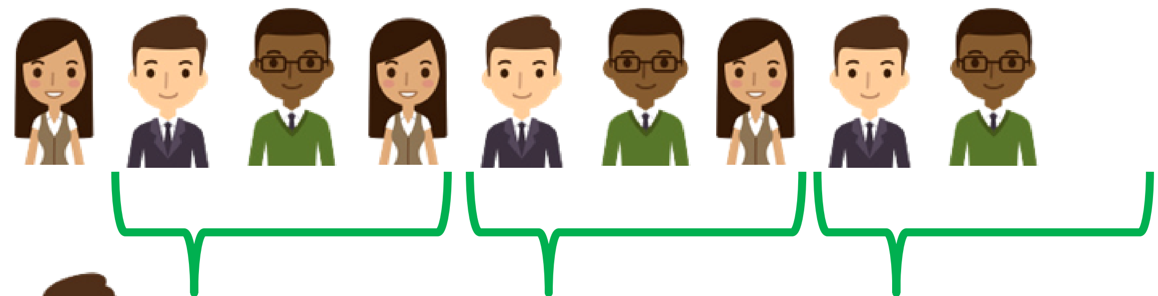
	$1200	$200	**$300**	$200	$100
	**$800**	$500	$200	**$300**	$200
	$800	**$400**	$400	$300	$100

The draft mechanism

- Drafting order: 

			ROLLS ROYCE		
	$1200	$200	**$300**	$200	**$100**
	**$800**	$500	$200	**$300**	$200
	$800	**$400**	$400	$300	$100

The draft mechanism

- Drafting order: 
- **Phases** for agent 
- In each phase,  prefers the good he gets to the good every other agent gets
- So, ignoring the good picked by an agent at the very beginning of the sequence,  is EF

Local search

- **Allocate goods one by one**
- In each step j:
  - Allocate good j **to an agent that nobody envies**
  - If this creates a "cycle of envy", **redistribute the bundles along the cycle**
- Crucial property:
  - Envy can be eliminated by removing just a **single good**
  - Implies **EF1**
- Lipton, Markakis, Mossel, & Saberi (2004)

Adding an efficiency objective

- **Pareto optimality (PO)**:
  - No alternative allocation exists that makes some agent better off without making any agents worse off
  - An allocation $A = (A_1, A_2, \ldots, A_n)$ is called **Pareto-optimal** if there is no allocation $B = (B_1, B_2, \ldots, B_n)$ such that $v_i(B_i) \geq v_i(A_i)$ for every agent i and $v_{i'}(B_{i'}) > v_{i'}(A_{i'})$ for some agent i'
- Easy to achieve: give each good to the agent that values it the most

EF1+PO?

EF1+PO?

- **Maximum Nash welfare (MNW)** allocation:
  – the allocation that maximizes the Nash welfare (**product of agent valuations**)
- **Theorem**: the MNW solution is EF1 and PO
  – C., Kurokawa, Moulin, Procaccia, Shah, & Wang (2016)

Theorem: MNW solution is EF1+PO

Theorem: MNW solution is EF1+**PO**

- **PO** is trivial since MNW maximizes $\prod_{i \in N} v_i(A_i)$

Theorem: MNW solution is **EF1**+PO

- Assume MNW is not EF1

Theorem: MNW solution is **EF1**+PO

- Assume MNW is not EF1
- Agent i envies agent j even after any single good is removed from j's bundle

Theorem: MNW solution is **EF1**+PO

- Assume MNW is not EF1
- Agent i envies agent j even after any single good is removed from j's bundle
- For good $g^* = \underset{g \in A_j : v_i(g) > 0}{\operatorname{argmin}} \frac{v_j(g)}{v_i(g)}$

  we have $\boxed{v_i(A_i) < v_i(A_j) - v_i(g^*)}$

Theorem: MNW solution is **EF1**+PO

- Recall that $g^* = \operatorname*{argmin}_{g \in A_j : v_i(g) > 0} \frac{v_j(g)}{v_i(g)}$

Theorem: MNW solution is **EF1**+PO

- Recall that $g^* = \underset{g \in A_j : v_i(g) > 0}{\operatorname{argmin}} \frac{v_j(g)}{v_i(g)}$

$$v_j(A_j) \geq \sum_{g \in A_j : v_i(g) > 0} v_j(g)$$

Theorem: MNW solution is **EF1**+PO

- Recall that $g^* = \operatorname*{argmin}_{g \in A_j : v_i(g) > 0} \frac{v_j(g)}{v_i(g)}$

$$v_j(A_j) \geq \sum_{g \in A_j : v_i(g) > 0} v_j(g) \geq \sum_{g \in A_j : v_i(g) > 0} v_i(g) \frac{v_j(g^*)}{v_i(g^*)}$$

Theorem: MNW solution is **EF1**+PO

- Recall that $g^* = \underset{g \in A_j : v_i(g) > 0}{\operatorname{argmin}} \frac{v_j(g)}{v_i(g)}$

$$v_j(A_j) \geq \sum_{g \in A_j : v_i(g) > 0} v_j(g) \geq \sum_{g \in A_j : v_i(g) > 0} v_i(g) \frac{v_j(g^*)}{v_i(g^*)}$$

$$= \frac{v_j(g^*)}{v_i(g^*)} \sum_{g \in A_j : v_i(g) > 0} v_i(g)$$

Theorem: MNW solution is EF1+PO

- Recall that $g^* = \underset{g \in A_j : v_i(g) > 0}{\operatorname{argmin}} \dfrac{v_j(g)}{v_i(g)}$

$$v_j(A_j) \geq \sum_{g \in A_j : v_i(g) > 0} v_j(g) \geq \sum_{g \in A_j : v_i(g) > 0} v_i(g) \frac{v_j(g^*)}{v_i(g^*)}$$

$$= \frac{v_j(g^*)}{v_i(g^*)} \sum_{g \in A_j : v_i(g) > 0} v_i(g) = \frac{v_j(g^*)}{v_i(g^*)} v_i(A_j)$$

Theorem: MNW solution is **EF1**+PO

- Recall that $g^* = \underset{g \in A_j : v_i(g) > 0}{\operatorname{argmin}} \frac{v_j(g)}{v_i(g)}$

$$v_j(A_j) \geq \sum_{g \in A_j : v_i(g) > 0} v_j(g) \geq \sum_{g \in A_j : v_i(g) > 0} v_i(g) \frac{v_j(g^*)}{v_i(g^*)}$$

$$= \frac{v_j(g^*)}{v_i(g^*)} \sum_{g \in A_j : v_i(g) > 0} v_i(g) = \frac{v_j(g^*)}{v_i(g^*)} v_i(A_j)$$

- Hence, $\boxed{v_i(g^*) v_j(A_j) - v_j(g^*) v_i(A_j) \geq 0}$

Theorem: MNW solution is **EF1**+PO

$$\boxed{v_i(A_i) < v_i(A_j) - v_i(g^*)}$$

$$\boxed{v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j) \geq 0}$$

Theorem: MNW solution is **EF1**+PO

$\boxed{v_i(A_i) < v_i(A_j) - v_i(g^*)}$ $\boxed{v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j) \geq 0}$

$v_i(A_i)\ v_j(A_j)$

Theorem: MNW solution is **EF1**+PO

$\boxed{v_i(A_i) < v_i(A_j) - v_i(g^*)}$ $\boxed{v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j) \geq 0}$

$$v_i(A_i)\, v_j(A_j)$$

$$\leq v_i(A_i)v_j(A_j) + v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j)$$

Theorem: MNW solution is **EF1**+PO

$\boxed{v_i(A_i) < v_i(A_j) - v_i(g^*)}$ $\boxed{v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j) \geq 0}$

$$v_i(A_i)\ v_j(A_j)$$

$$\leq v_i(A_i)v_j(A_j) + v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j)$$

$$< v_i(A_i)v_j(A_j) + v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_i) - v_j(g^*)v_i(g^*)$$

Theorem: MNW solution is EF1+PO

$\boxed{v_i(A_i) < v_i(A_j) - v_i(g^*)}$ $\boxed{v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j) \geq 0}$

$$
\begin{aligned}
& v_i(A_i)\, v_j(A_j) \\
& \leq v_i(A_i)v_j(A_j) + v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j) \\
& < v_i(A_i)v_j(A_j) + v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_i) - v_j(g^*)v_i(g^*) \\
& = (v_i(A_i) + v_i(g^*)) \cdot (v_j(A_j) - v_j(g^*))
\end{aligned}
$$

Theorem: MNW solution is **EF1**+PO

$\boxed{v_i(A_i) < v_i(A_j) - v_i(g^*)}$ $\boxed{v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j) \geq 0}$

$$v_i(A_i)\, v_j(A_j)$$

$$\leq v_i(A_i)v_j(A_j) + v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_j)$$

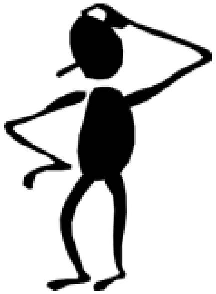
$$< v_i(A_i)v_j(A_j) + v_i(g^*)v_j(A_j) - v_j(g^*)v_i(A_i) - v_j(g^*)v_i(g^*)$$

$$= (v_i(A_i) + v_i(g^*)) \cdot (v_j(A_j) - v_j(g^*))$$

- So A is not a MNW solution, a contradiction.

- **QED**

Computational issues



- **EF1**+**PO** in **polynomial time**?
  - Yes for two agents (using a restricted MNW solution)
  - Open for more agents (e.g., three agents)
  - Several attempts (e.g., rounding a fractional MNW solution) miserably failed
  - Some progress in very recent work by Barman, Murthy, & Vaish (2018)

Summary

- What have we covered today?
  - Maxmin fair share
  - EF1
  - Draft mechanism
  - Local search
  - Max Nash Welfare mechanism