Algorithms for Fair Allocation

XIAOWEI WU

IOTSC, UNIVERSITY OF MACAU

Disclaimer

In this tutorial, we will NOT

- Assume any prior knowledge of fair allocation problems
- Walk you through tedious, detailed proofs
- Claim to present a complete overview of the entire fair allocation realm

Instead, we will introduce

- What is the fair allocation problem
- What are the popular fairness measurements
- Some recent results and algorithms

Outline

- Fair Allocation of Indivisible Items
- Fairness Notions and Relaxations
- Algorithms for Computing Fair Allocations
- Other Settings & Extensions

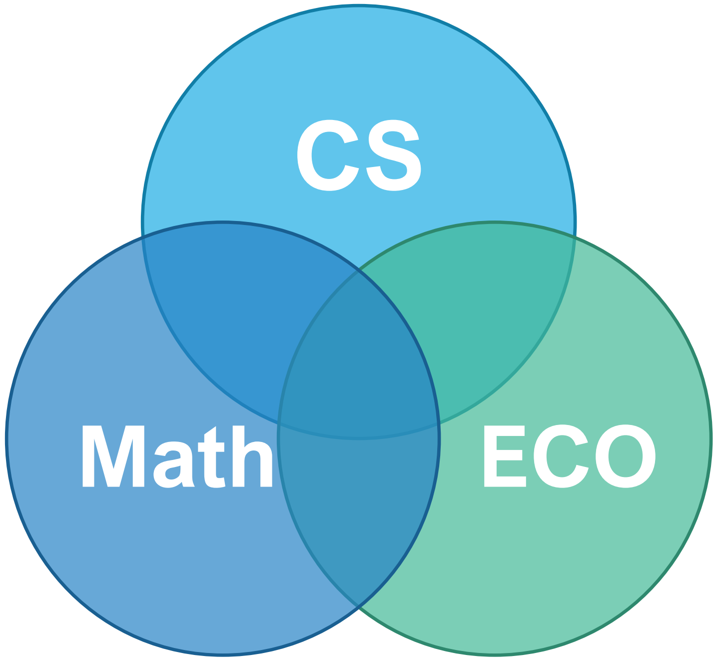
The Study on Fair Allocations

Two main problems:

- to measure fairness
- to compute fair allocations

Research area that intersects with

- Computer Science
- Mathematics
- Economics



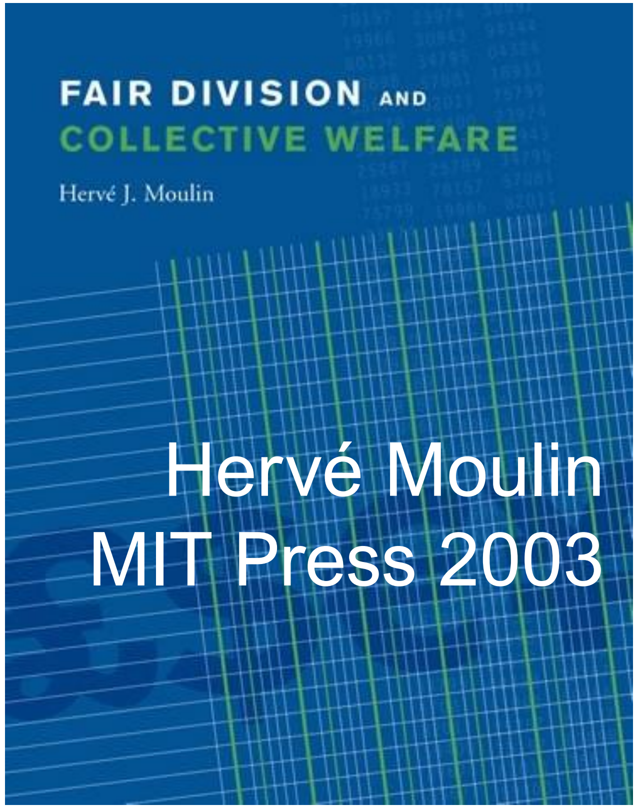
The Study on Fair Allocations

Two main problems:

- to measure fairness
- to compute fair allocations

Research area that intersects with

- Computer Science
- Mathematics
- Economics



Fair Allocations

Cake cutting problem [Steinhaus, Econ 1948]

Agents $N = \{1,2,\dots,n\}$

- Each agent $i \in N$ has a valuation function v_i

Rules:

- Full allocation
- Arbitrarily partition (divisible)
- Envy-free: no agent envies another agent

Divisible vs. Indivisible

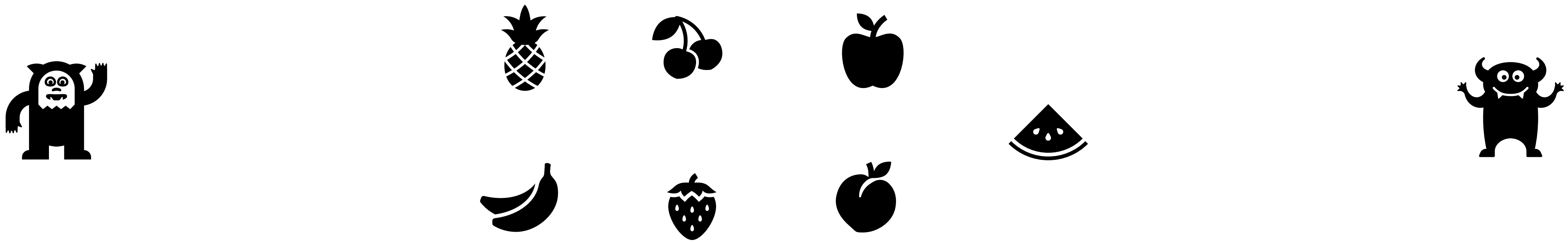
Divisible items

Indivisible items

Allocation of Indivisible Items

A set of indivisible items $M = \{1,2,\ldots,m\}$ and a group of agents $N = \{1,2,\ldots,n\}$

Task: allocate the items to the agents



Allocation of Indivisible Items

A set of indivisible items $M = \{1,2,\dots,m\}$ and a group of agents $N = \{1,2,\dots,n\}$

Task: allocate the items to the agents

Allocation of Indivisible Items

A set of indivisible items $M = \{1,2,\ldots,m\}$ and a group of agents $N = \{1,2,\ldots,n\}$

Task: allocate the items to the agents

Different agents may have different values on the items

Allocation of Indivisible Items

A set of indivisible items $M = \{1,2, \dots, m\}$ and a group of agents $N = \{1,2, \dots, n\}$

Task: allocate the items to the agents

Each agent $i \in N$ has value / utility $v_i(e) \geq 0$ on item $e \in M$: (additive valuation function)

- Valuation function of agent i : $v_i(X) = \sum_{e \in X} v_i(e)$, for $X \subseteq M$

Allocation of Indivisible Items

A set of indivisible items $M = \{1,2,\ldots,m\}$ and a group of agents $N = \{1,2,\ldots,n\}$

Task: allocate the items to the agents

Each agent $i \in N$ has value / utility $v_i(e) \geq 0$ on item $e \in M$: (additive valuation function)

- Valuation function of agent i : $v_i(X) = \sum_{e\in X} v_i(e)$, for $X \subseteq M$

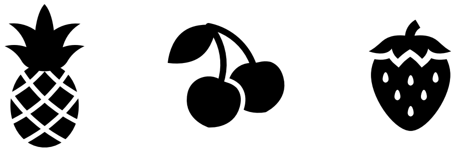
An **allocation** $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ is a **partition of** $\boldsymbol{M}$ **into** $\boldsymbol{n}$ **bundles**

- $\bigcup_{i\in N} X_i = M$ and $X_i \cap X_j = \emptyset$ for all $i \neq j$
- Agent i receives bundle X_i, and has utility $v_i(X_i)$

Allocation of Indivisible Goods / Chores

Allocation of **goods** :

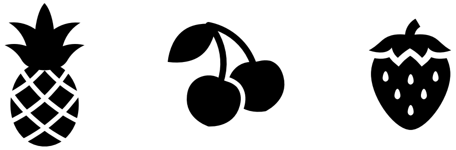
- Each agent i has value $v_i(e) > 0$ on item e
- Agents would like to maximize their own values



Allocation of Indivisible Goods / Chores

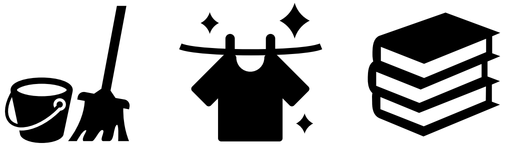
Allocation of **goods** :

- Each agent i has value $v_i(e) > 0$ on item e
- Agents would like to maximize their own values



Allocation of **chores** :

- Each agent i has cost $c_i(e) > 0$ on item e
- Agents would like to minimize their own costs



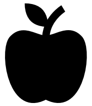
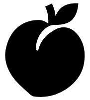
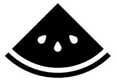
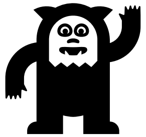
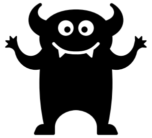
Social Welfare of Allocations

Social Welfare of allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$: $\mathrm{SW}(\boldsymbol{X}) = \sum_{i \in N} v_i(X_i)$

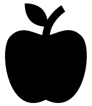
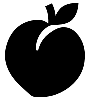
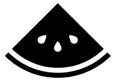
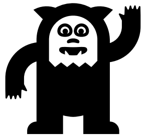
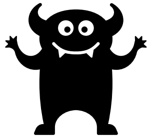
To maximize social welfare:

allocate each item to the agent with maximum value on the item

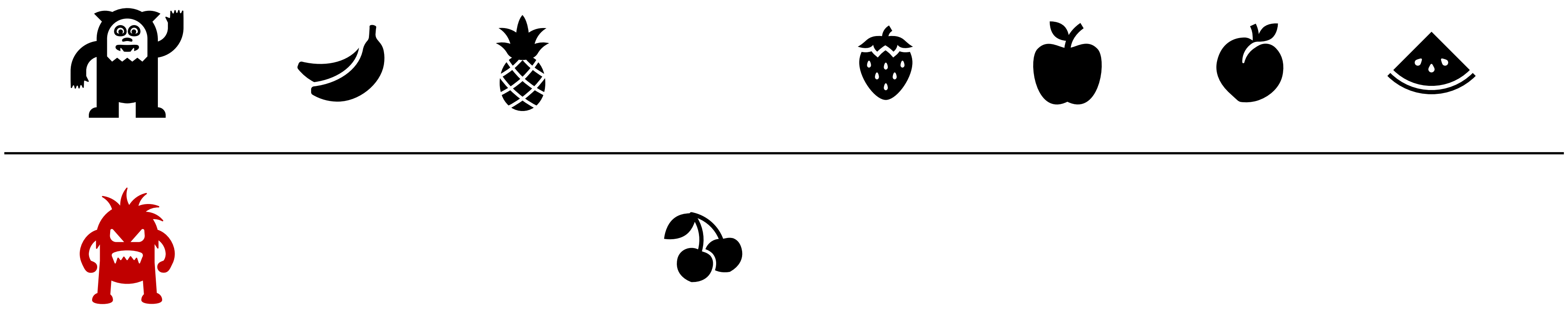
Example

							
	3	15	4	5	8	10	12
	2	10	5	3	5	7	8

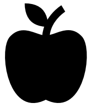
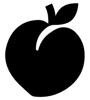
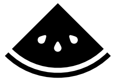
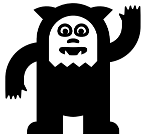
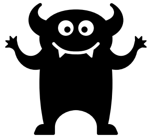
Example

							
	**3**	**15**	4	**5**	**8**	**10**	**12**
	2	10	**5**	3	5	7	8

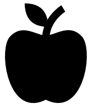
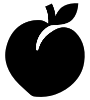
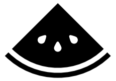
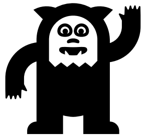
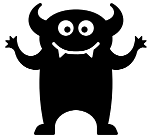
Example



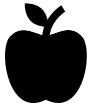
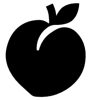
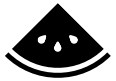
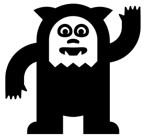
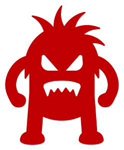
Example

							
	3	15	4	5	8	10	12
	2	10	5	3	5	7	8

Normalized Valuations

							
	0.053	0.263	0.07	0.088	0.14	0.176	0.21
	0.05	0.25	0.125	0.075	0.125	0.175	0.2

Normalized Valuations

							
	**0.053**	**0.263**	0.07	**0.088**	**0.14**	**0.176**	**0.21**
	0.05	0.25	**0.125**	0.075	0.125	0.175	0.2

Nash Social Welfare (NSW)

Nash Social Welfare

of allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$

: $\mathrm{NSW}(\boldsymbol{X}) = \left(\Pi_{i \in N} v_i(X_i)\right)^{1/n}$

Nash Social Welfare (NSW)

Nash Social Welfare of allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$: $\mathrm{NSW}(\boldsymbol{X}) = \Pi_{i \in N} v_i(X_i)$

Example

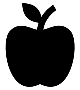
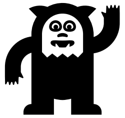
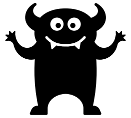
	3	15	4	5	8	10	12
	2	10	5	3	5	7	8

Example

	3	**15**	4	**5**	**8**	**10**	**12**
	2	10	**5**	3	5	7	8

NSW = 5 × 53 = 265

Example

							
	3	**15**	4	**5**	**8**	10	12
	**2**	10	**5**	3	5	**7**	**8**

NSW = 22 × 28 = 616

Fairness Notions

Fairness Notions

PROP

Proportional

EF

Envy-Free

MMS

Maximin Share

and their relaxations and approximations

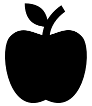
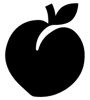
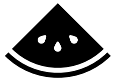
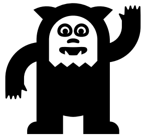
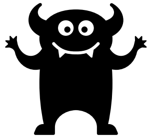
Proportional Allocations

Proportional (PROP)

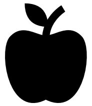
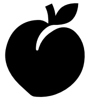
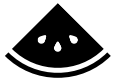
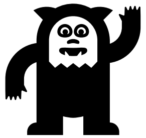
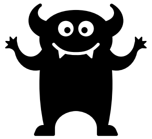
Allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ has

$$\forall i \in N: \; v_i(X_i) \geq 1/n \cdot v_i(M)$$

Example

							
	0.053	0.263	0.07	0.088	0.14	0.176	0.21
	0.05	0.25	0.125	0.075	0.125	0.175	0.2

Example

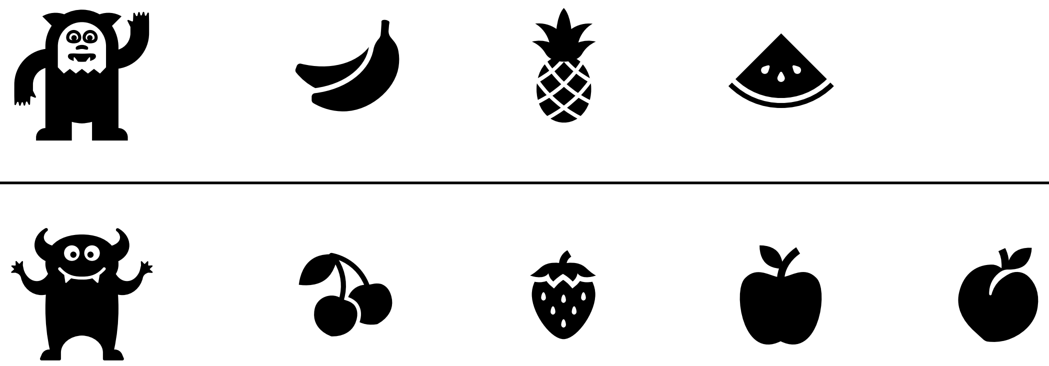
							
	**0.053**	**0.263**	0.07	0.088	0.14	0.176	**0.21**
	0.05	0.25	**0.125**	**0.075**	**0.125**	**0.175**	0.2

Example

	Utility
	0.526
	0.5

Example

The allocation
is **PROP**



	Utility
Row 1	0.526
Row 2	0.5

Proportional Allocations

Proportional (PROP)

Allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ has

$$\forall i \in N : \ v_i(X_i) \geq 1/n \cdot v_i(M)$$

For allocation of **indivisible** items,

PROP allocations do not always exist

Proportional Allocations

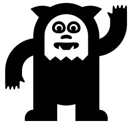
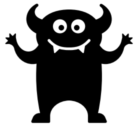
Proportional (PROP)

Allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$ has

$$\forall i \in N : \; v_i(X_i) \geq 1/n \cdot v_i(M)$$

For allocation of **indivisible** items,

PROP allocations do not always exist

	
	1
	1

Envy

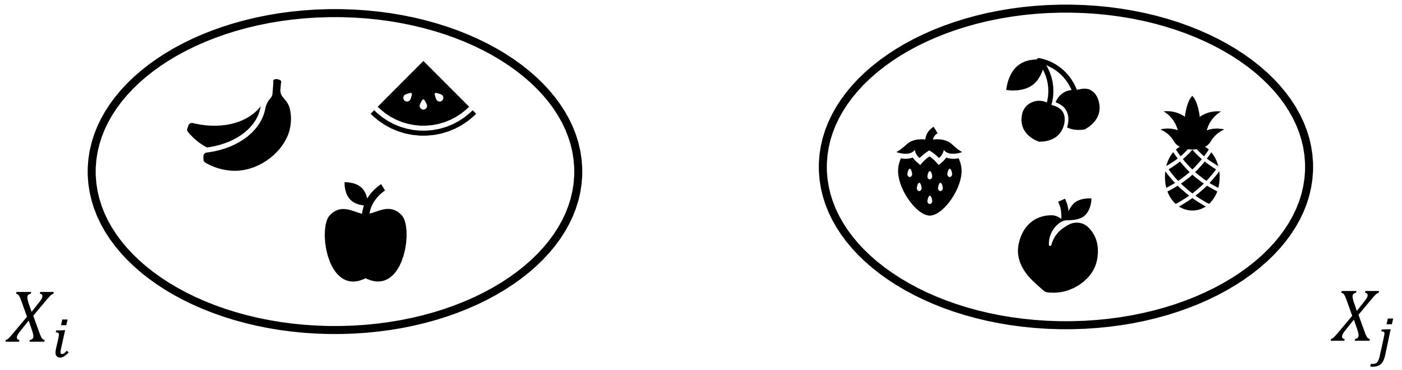
Given allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$, agent i **envies** agent j if

$$v_i(X_i) < v_i(X_j)$$

Envy

Given allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$, agent i **envies** agent j if

$$v_i(X_i) < v_i(X_j)$$

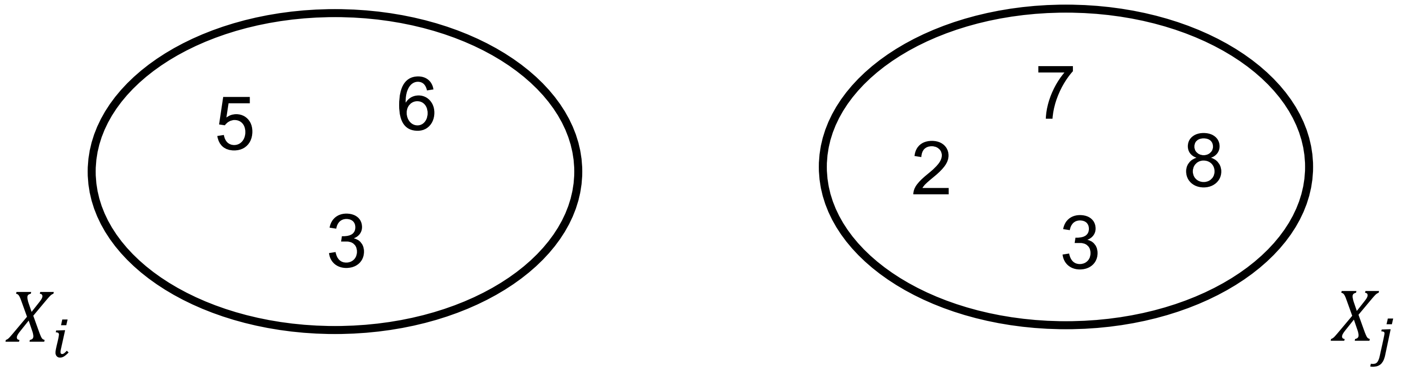


Envy

Given allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$, agent i **envies** agent j if

$$v_i(X_i) < v_i(X_j)$$

Under valuation function v_i:

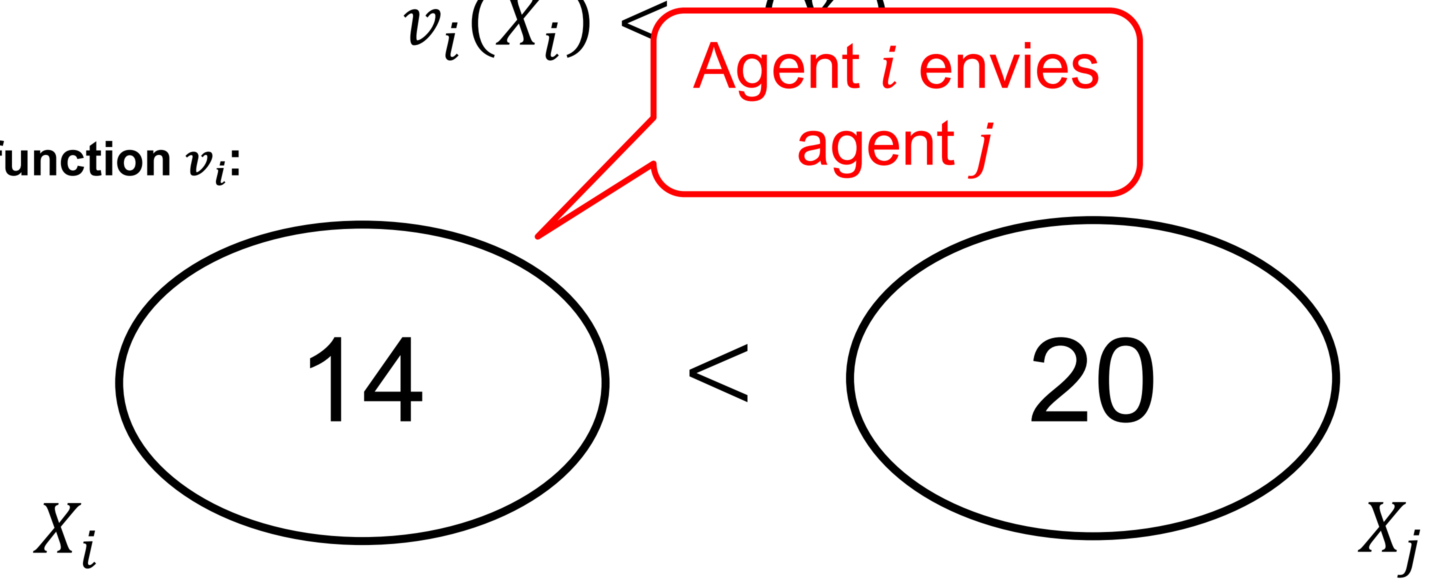


Envy

Given allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$, agent i **envies** agent j if

$$v_i(X_i) <$$

Under valuation function v_i:



Envy-Free Allocations

Given allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$, agent i **envies** agent j if

$$v_i(X_i) < v_i(X_j)$$

Allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$ is **envy-free** if no agent envies another agent, i.e.,

$$\forall i, j \in N, \qquad v_i(X_i) \geq v_i(X_j)$$

EF Allocations Hardly Exist

EF allocations are not guaranteed to exist, even under identical valuations

EF Allocations Hardly Exist

EF allocations are not guaranteed to exist, even under identical valuations

Observation. If agent i does not envy any other agent, then $v_i(X_i) \geq 1/n \cdot v_i(M)$

$$\sum_{j \in N} v_i(X_i) \geq \sum_{j \in N} v_i(X_j)$$

EF Allocations Hardly Exist

EF allocations are not guaranteed to exist, even under identical valuations

Observation. If agent i does not envy any other agent, then $v_i(X_i) \geq 1/n \cdot v_i(M)$

$$n \cdot v_i(X_i) = \sum_{j \in N} v_i(X_i) \geq \sum_{j \in N} v_i\left(X_j\right) = v_i(M)$$

EF Allocations Hardly Exist

EF allocations are not guaranteed to exist, even under identical valuations

Observation. If agent i does not envy any other agent, then $v_i(X_i) \geq 1/n \cdot v_i(M)$

Lemma. Every EF allocation is PROP.

EF $\Rightarrow$ PROP

Relaxations of Envy-Freeness

Envy-free up to one item (**EF1**) :

“The envy between two agents can be eliminated after removing some item.”

Relaxations of Envy-Freeness

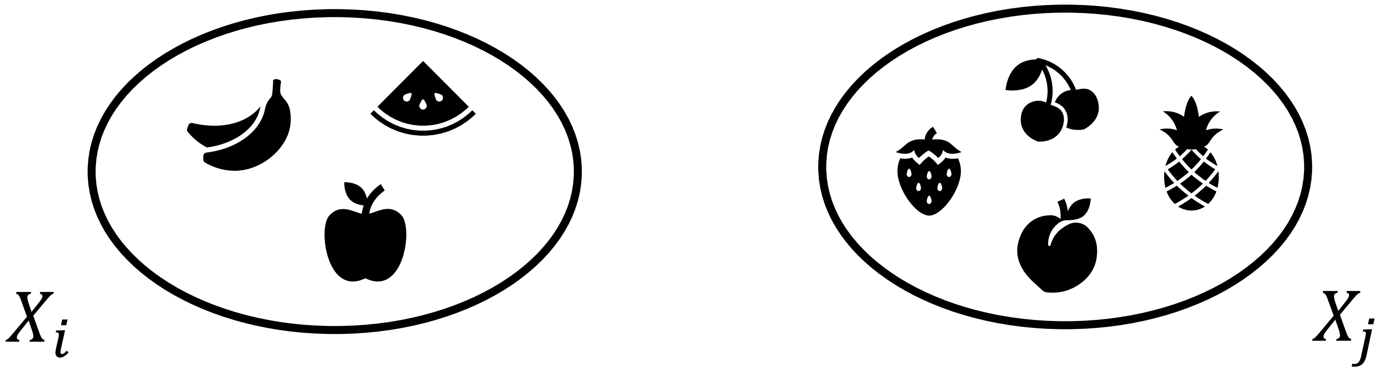
Envy-free up to one item (**EF1**) : $\forall i \in N, \forall j \in N$:

$$\exists e \in X_j : v_i(X_i) \geq v_i\left(X_j \setminus \{e\}\right)$$

Relaxations of Envy-Freeness

Envy-free up to one item (**EF1**) : $\forall i \in N, \forall j \in N$:

$$\exists e \in X_j : v_i(X_i) \geq v_i(X_j \setminus \{e\})$$

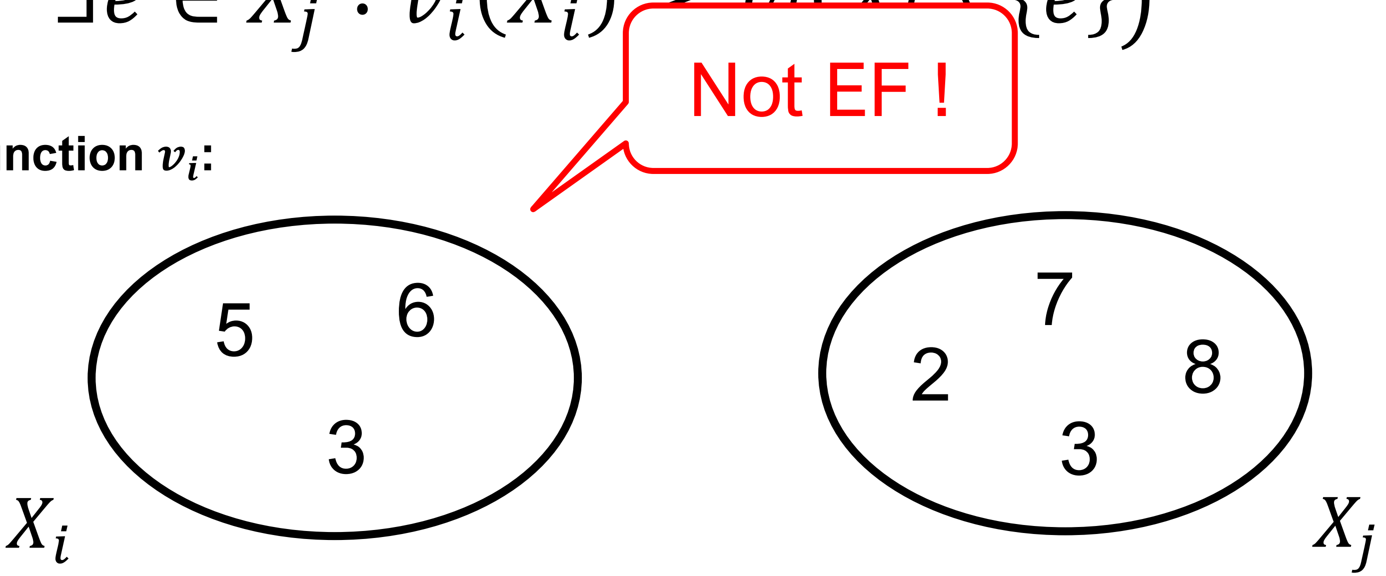


Relaxations of Envy-Freeness

Envy-free up to one item (**EF1**) : $\forall i \in N, \forall j \in N$:

$$\exists e \in X_j \ : v_i(X_i) \geq v_i(X_j \setminus \{e\})$$

Under valuation function v_i:

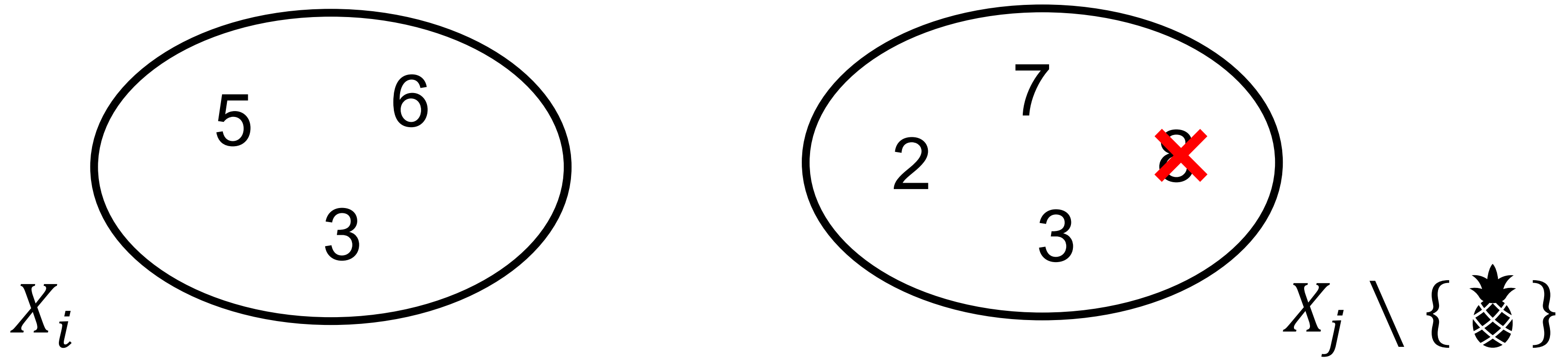


Relaxations of Envy-Freeness

Envy-free up to one item (**EF1**) : $\forall i \in N, \forall j \in N$:

$$\exists e \in X_j : v_i(X_i) \geq v_i\left(X_j \setminus \{e\}\right)$$

Under valuation function v_i:

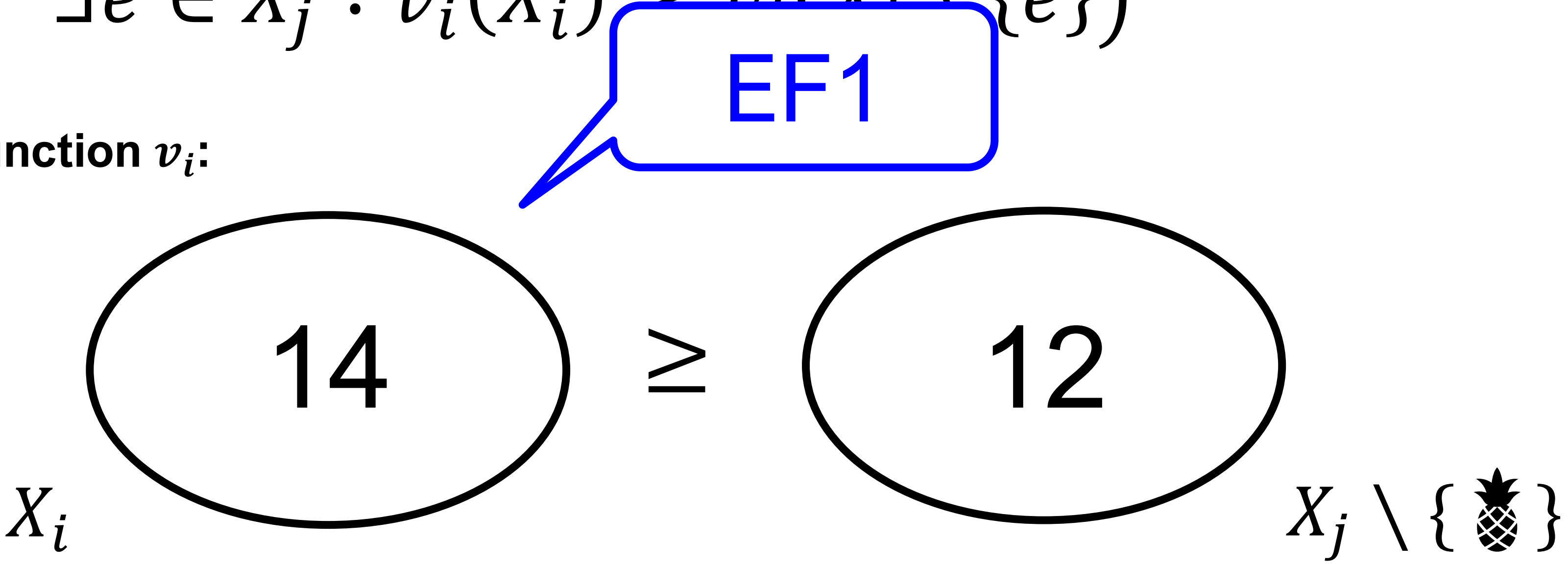


Relaxations of Envy-Freeness

Envy-free up to one item (**EF1**) : $\forall i \in N, \forall j \in N$:

$$\exists e \in X_j : v_i(X_i) \geq v_i(X_j \setminus \{e\})$$

Under valuation function v_i:



Relaxations of Envy-Freeness

Envy-free up to one item (**EF1**) : $\forall i \in N, \forall j \in N$:

$$\exists e \in X_j : v_i(X_i) \geq v_i\left(X_j \setminus \{e\}\right)$$

*"By removing **some** item, agent i does not envy agent j"*

Relaxations of Envy-Freeness

Envy-free up to any item (**EFX**) :

"The envy between two agents can be eliminated after removing any item."

Relaxations of Envy-Freeness

Envy-free up to any item (**EFX**) : $\forall i \in N, \forall j \in N$:

$$\forall e \in X_j : v_i(X_i) \geq v_i(X_j \setminus \{e\})$$

Relaxations of Envy-Freeness

Envy-free up to any item (**EFX**) : $\forall i \in N$, $\forall j \in N$:

$$\forall e \in X_j : v_i(X_i) \geq v_i\left(X_j \setminus \{e\}\right)$$

Stronger fairness requirement than EF1.

"By removing ***any*** *item from X_j, agent i does not envy agent j"*

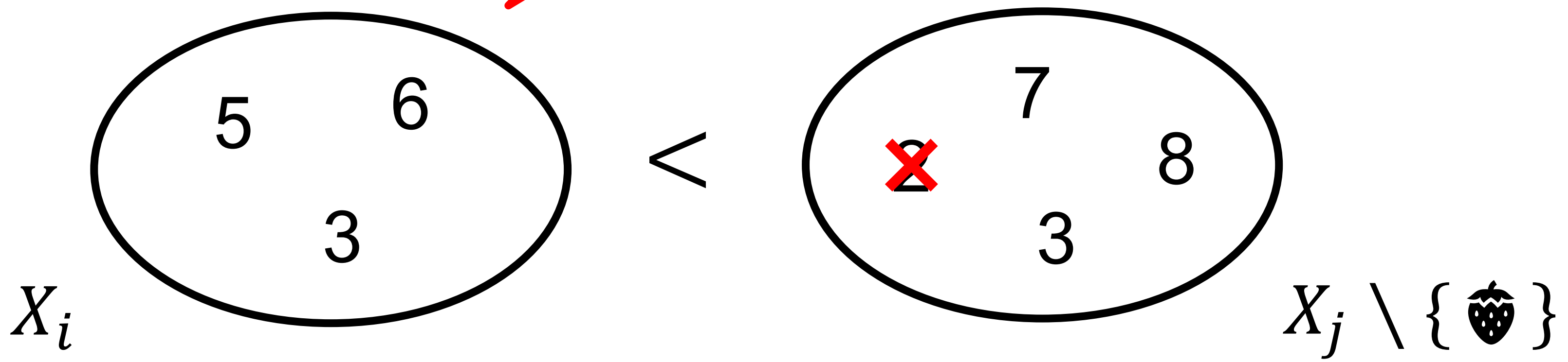
Relaxations of Envy-Freeness

Envy-free up to any item (**EFX**) : $\forall i \in N$, $\forall j \in N$:

$$\forall e \in X_j : v_i(X_i) \geq v_i(X_j \setminus \{e\})$$

not EFX !

Under valuation function v_i:



Relaxations of Envy-Freeness

Envy-free up to any item (**EFX**) : $\forall i \in N, \forall j \in N$:

$$\forall e \in X_j : v_i(X_i) \geq v_i(X_j \setminus \{e\})$$

Envy-free up to one item (**EF1**) : $\forall i \in N, \forall j \in N$:

$$\exists e \in X_j : v_i(X_i) \geq v_i(X_j \setminus \{e\})$$

EF ⇒ EFX ⇒ EF1

Relaxations of Proportionality

Proportional up to any item (**PROPX**) : $\forall i \in N$:

$$\forall e \notin X_i : v_i(X_i \cup \{e\}) \geq 1/n \cdot v_i(M)$$

Proportional up to one item (**PROP1**) : $\forall i \in N$:

$$\exists e \notin X_i : v_i(X_i \cup \{e\}) \geq 1/n \cdot v_i(M)$$

PROP ⇒ PROPX ⇒ PROP1

Fairness Notions

Comparison Based:

- EF, EFX, EF1

Threshold Based:

- PROP, PROPX, PROP1

Fairness Notions

Comparison Based:

- EF, EFX, EF1

Threshold Based:

- PROP, PROPX, PROP1
- MMS

Maximin Share (MMS)

Maximin Share (MMS) of agent $i \in N$ [Budish, JPE 2011]

Suppose agent i partitions the items M into n bundles and lets the other $n-1$ agents pick bundles first: i should try to maximize the worst bundle ($\min_{j \in N}\{v_i(X_j)\}$)

Maximin Share (MMS)

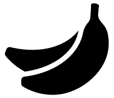
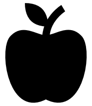
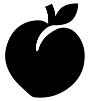
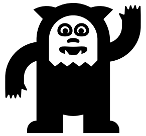
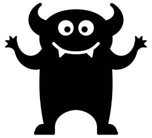
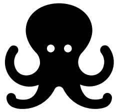
Maximin Share (MMS) of agent $i \in N$ [Budish, JPE 2011]

Suppose agent i partitions the items M into n bundles and lets the other $n-1$ agents pick bundles first: i should try to maximize the worst bundle ($\min_{j \in N}\{v_i(X_j)\}$)

Let $\Pi_n(M)$ be the set of all n-partitions of items in M:

$$\mathrm{MMS}_i(M, n) = \max_{(X_1, \ldots, X_n) \in \Pi_n(M)} \min_{j \in N}\{v_i(X_j)\}$$

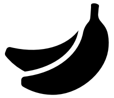
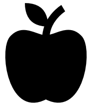
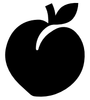
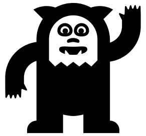
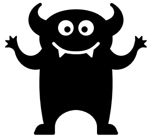
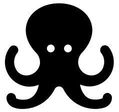
Example

							**MMS**
	3	15	4	5	8	10	
	2	11	5	3	5	7	
	6	11	1	4	7	9	

Example

							MMS
	3	15	4	5	8	10	
	2	11	5	3	5	7	
	6	11	1	4	7	9	

Example

							MMS
	3	15	4	5	8	10	
	2	11	5	3	5	7	
	6	11	1	4	7	9	

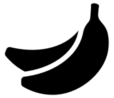
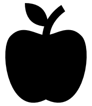
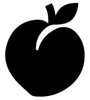
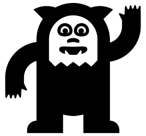
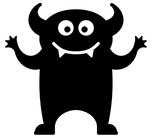
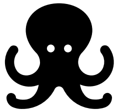
Example

							MMS
	3	15	4	5	8	10	
	2	11	5	3	5	7	
	6	11	1	4	7	9	

Example

							MMS
	3	15	4	5	8	10	15
	2	11	5	3	5	7	
	6	11	1	4	7	9	

Example

							**MMS**
	3	15	4	5	8	10	**15**
	2	11	5	3	5	7	**10**
	6	11	1	4	7	9	

Example

							MMS
	3	15	4	5	8	10	**15**
	2	11	5	3	5	7	**10**
	6	11	1	4	7	9	**12**

MMS Fair Allocation

Maximin Share (MMS) of agent $i \in N$

Let $\Pi_n(M)$ be the set of all n-partitions of items in M:

$$\mathrm{MMS}_i(M,n) = \max_{(X_1,\ldots,X_n)\in\Pi_\mathrm{n}(M)} \min_{j\in N}\{v_i(X_j)\}$$

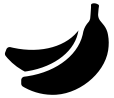
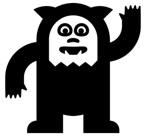
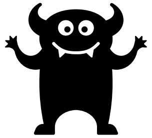
- $\mathrm{MMS}_i \le \mathrm{PROP}_i = 1/n \cdot v_i(M)$

An allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ is **MMS** if $v_i(X_i) \ge \mathrm{MMS}_i$ for all $i \in N$

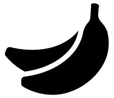
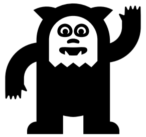
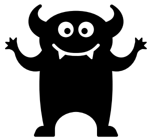
Example

							MMS
	3	15	4	5	8	10	15
	2	11	5	3	5	7	10
	6	11	1	4	7	9	12

Example

		MMS
	3	?
	2	?

Example

		MMS
	3	**0**
	2	**0**

Existence and Computation of Fair Allocations

Non-existence

PROP allocations are not guaranteed to exist

EF allocations are not guaranteed to exist

Non-existence

PROP allocations are not guaranteed to exist

EF allocations are not guaranteed to exist

MMS allocations are not guaranteed to exist

- For goods [KurokawaPW, JACM 2018], for chores [AzizBLM, AAAI 2017]

Relaxations and Approximations?

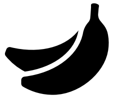
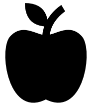
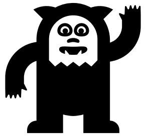
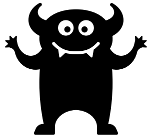
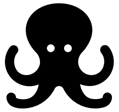
PROP1/PROPX Allocations

PROP1 allocations always exist [AzizMS, ORL 2020]

- Even for mixture of goods and chores, and with Pareto-optimality guarantee

PROPX allocations are not guaranteed to exist [AzizMS, ORL 2020]

Example [AzizMS, ORL 2020]

					
	3	3	3	3	1
	3	3	3	3	1
	3	3	3	3	1

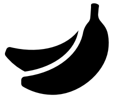
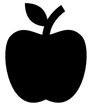
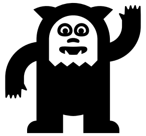
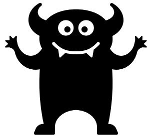
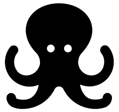
Example [AzizMS, ORL 2020]

						PROP
	3	3	3	3	1	**4.33**
	3	3	3	3	1	**4.33**
	3	3	3	3	1	**4.33**

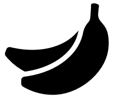
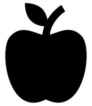
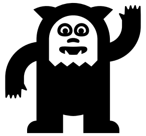
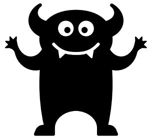
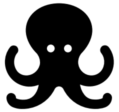
Example [AzizMS, ORL 2020]

						PROP
	3	3	3	3	1	4.33
	3	3	3	3	1	4.33
	3	3	3	3	1	4.33

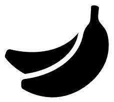
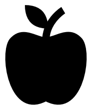
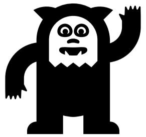
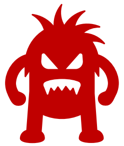
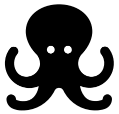
Example [AzizMS, ORL 2020]

						PROP
	3	3	3	**3**	**1**	**4.33**
	3	3	3	3	1	**4.33**
	3	3	3	3	1	**4.33**

Example [AzizMS, ORL 2020]

						PROP
	3	3	3	**3**	**1**	**4.33**
	**3**	3	3	3	1	**4.33**
	3	**3**	**3**	3	1	**4.33**

Example [AzizMS, ORL 2020]

						**PROP**
	3	3	3	**3**	**1**	**4.33**
	**3**	3	3	3	1	**4.33**
	3	**3**	**3**	3	1	**4.33**

PROP1/PROPX Allocations

PROP1 allocations always exist [AzizMS, ORL 2020]

- Even for mixture of goods and chores, and with Pareto-optimality guarantee

PROPX allocations are not guaranteed to exist [AzizMS, ORL 2020]

- In contrast, PROPX allocations always exist for chores [Moulin, ARE 2018; LiLW, WWW 2022]

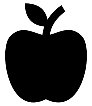
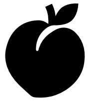
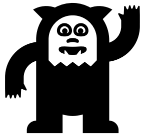
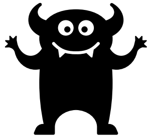
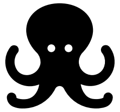
EF1 Allocations Always Exist

Round-Robin Algorithm [CaragiannisMPSW, TAEC 2019]

Repeat:

- For agent $i = 1,2, \dots, n$:
  - Let agent i pick her favourite unallocated item
  - **Until** all items are allocated

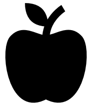
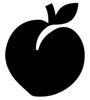
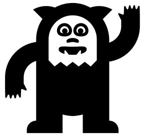
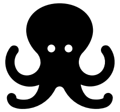
Round-Robin Allocation

						
	3	15	4	5	8	10
	2	11	5	3	5	7
	6	11	1	4	7	9

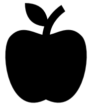
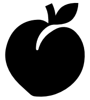
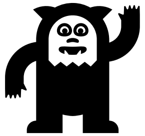
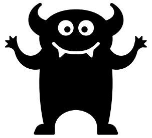
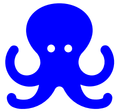
Round-Robin Allocation

	3	15	4	5	8	10
	2	11	5	3	5	7
	6	11	1	4	7	9

Round-Robin Allocation

						
	3	**15**	4	5	8	10
	2	11	5	3	5	**7**
	6	11	1	4	7	9

Round-Robin Allocation

						
	3	**15**	4	5	8	10
	2	11	5	3	5	**7**
	6	11	1	4	**7**	9

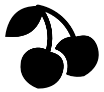
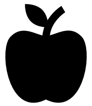
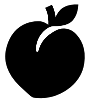
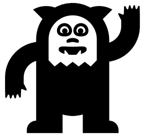
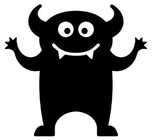
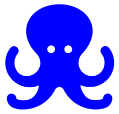
Round-Robin Allocation

	3	**15**	4	**5**	8	10
	2	11	5	3	5	**7**
	6	11	1	4	**7**	9

Round-Robin Allocation

	3	**15**	4	**5**	8	10
	2	11	**5**	3	5	**7**
	6	11	1	4	**7**	9

Round-Robin Allocation

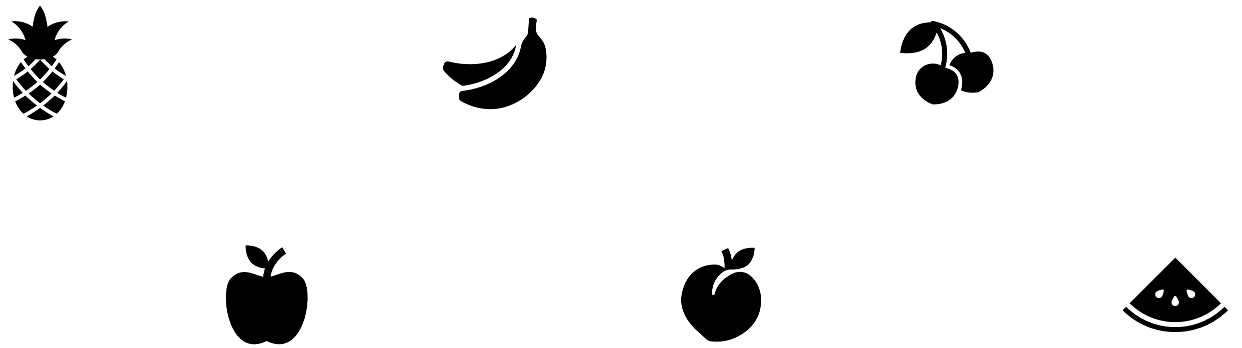
						
	3	**15**	4	**5**	8	10
	2	11	**5**	3	5	**7**
	**6**	11	1	4	**7**	9

The resulting allocation is EF1 because

Consider any agent $i \in N$ and agent $j \in N$:

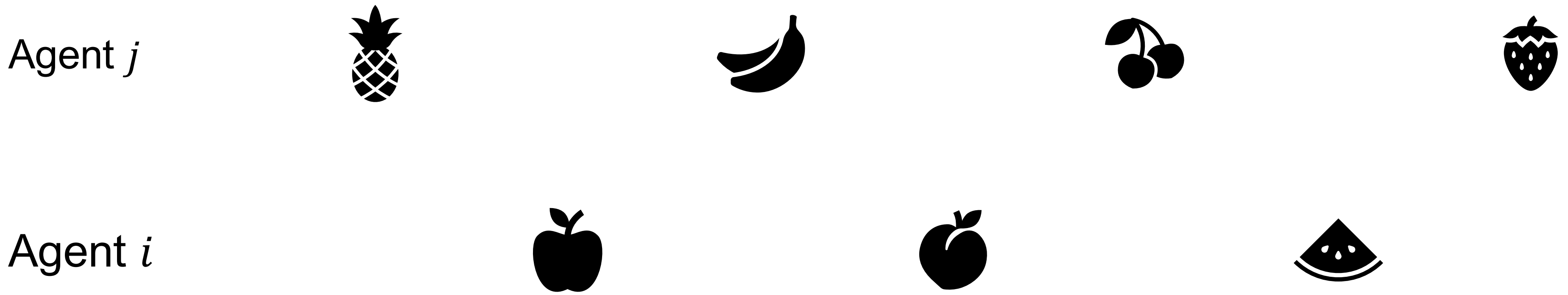
Agent j

Agent i



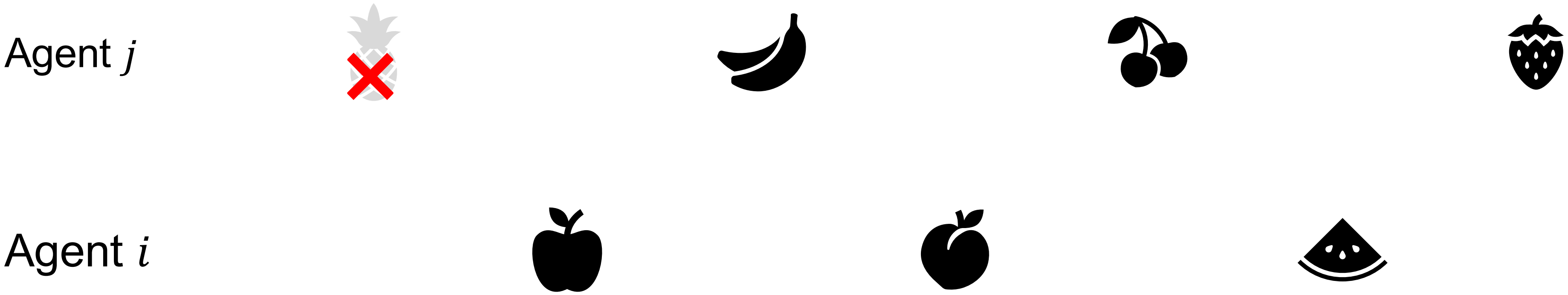
The resulting allocation is EF1 because

Consider any agent $i \in N$ and agent $j \in N$:
agent i does not envy agent j by more than one item because



The resulting allocation is EF1 because

Consider any agent $i \in N$ and agent $j \in N$:
agent i does not envy agent j by more than one item because

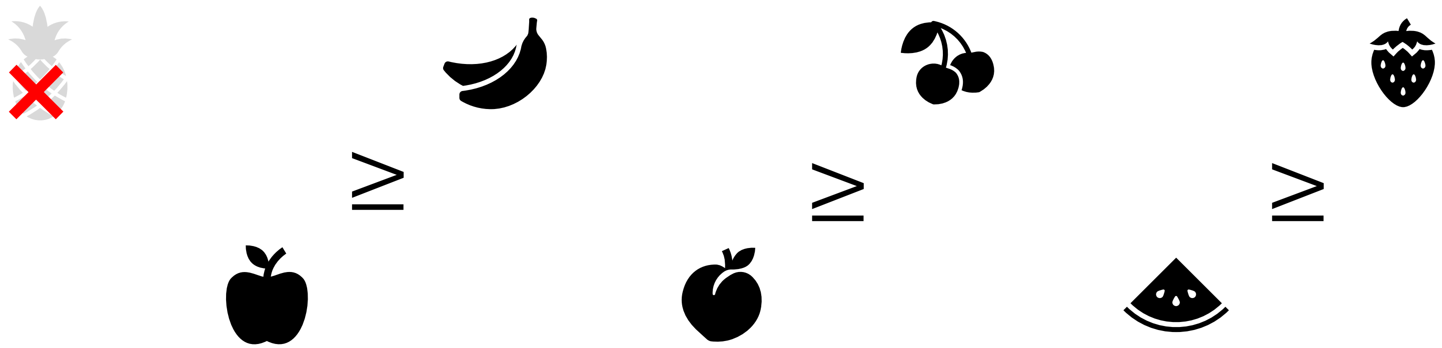


The resulting allocation is EF1 because

Consider any agent $i \in N$ and agent $j \in N$:
agent i does not envy agent j by more than one item because

Agent j

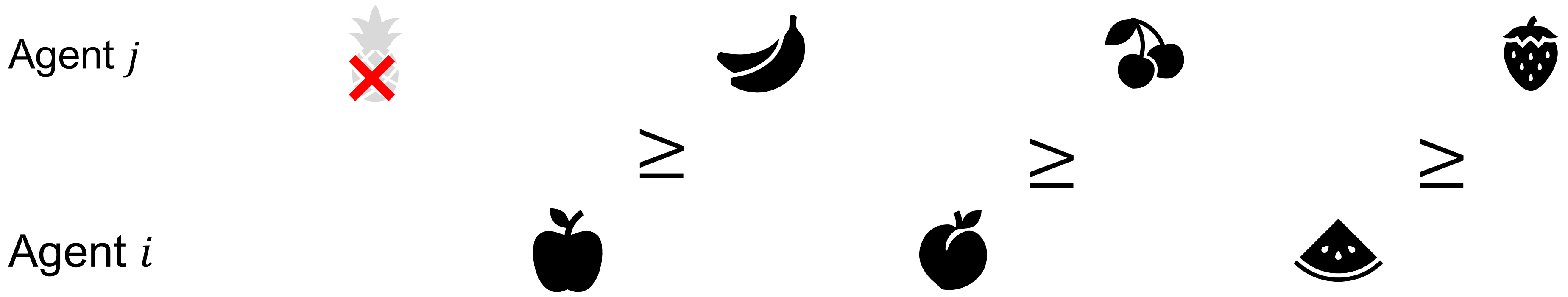
Agent i



The resulting allocation is EF1 because

Consider any agent $i \in N$ and agent $j \in N$:
agent i does not envy agent j by more than one item because

EF1 !



Extension: Sequential Picking Algorithms

Sequential Picking Algorithms

Fix a sequence of agents: $\sigma \in [n]^m$

- For agent $i = \sigma(1), \sigma(2), \dots, \sigma(m)$:
  - Let agent i pick her favourite unallocated item

Extension: Sequential Picking Algorithms

Sequential Picking Algorithms

Fix a sequence of agents: $\sigma \in [n]^k$, for some $k \geq n$

Repeat:

- For agent $i = \sigma(1), \sigma(2), \dots, \ \sigma(k)$:
  - Let agent i pick her favourite unallocated item
  - **Until** all items are allocated

Existence of EFX Allocations

[Plaut and Roughgarden, SIDMA 2020]

EFX allocation always exists for

- Agents with identical valuations
- Two-agents (with general valuations)
- Identical ordering (IDO) instances

Computation of EFX Allocations

For identical valuations: (Load Balancing)

Suppose $v(e_1) \geq v(e_2) \geq \cdots \geq v(e_m)$

Initialize $X_i \leftarrow \emptyset$ for all $i \in N$

for $t = 1,2, \ldots, m$:

- let $i \in N$ be the agent with minimum $v(X_i)$
- update $X_i \leftarrow X_i \cup \{e_t\}$

The resulting allocation is EFX because

Consider any agent $j \in N$ with bundle X_j

- Let $e_t \in M$ be the last item agent j receives
- For all $i \neq j$, we have $v(X_i) \geq v\left(X_j \setminus \{e_t\}\right)$
- For all $e \in X_j$, $v(e) \geq v(e_t)$

[**EFX**] For all agent $i, j \in N$ and $e \in X_j$, $v(X_i) \geq v\left(X_j \setminus \{e\}\right)$

Computation of EFX Allocations

For 2 agents (Divide-and-Choose):

Let agent 1 divide the items into two bundles Y_1 and Y_2

- by computing an EFX allocation based on v_1

Let agent 2 choose her preferred bundle, and leave the other bundle to agent 1

- $v_2(X_2) \geq v_2(X_1)$

EFX for both agents

Computation of EFX Allocations

Identical Ordering (IDO) instances:

Let $M = \{e_1, e_2, \ldots, e_m\}$. For all agent $i \in N$: $v_i(e_1) \geq v_i(e_2) \geq \cdots \geq v_i(e_m)$.

- All agents agree on the same ordering of items
- The values can still be different

An EFX allocation can be computed for every IDO instance

- using the envy-cycle elimination technique

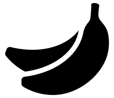
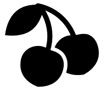
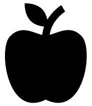
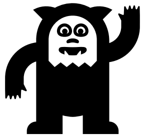
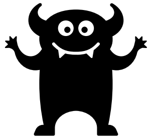
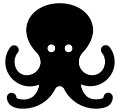
Computation of EFX Allocations

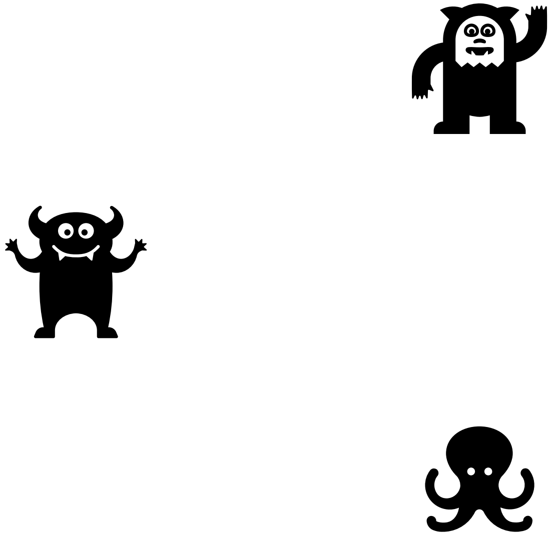
Envy-Cycle Elimination [LiptonMMS, EC 2004]

Envy-graph for a given (partial) allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$

- Directed graph $G(N, E)$: $(i, j) \in E$ if i envies j $(v_i(X_i) < v_i(X_j))$

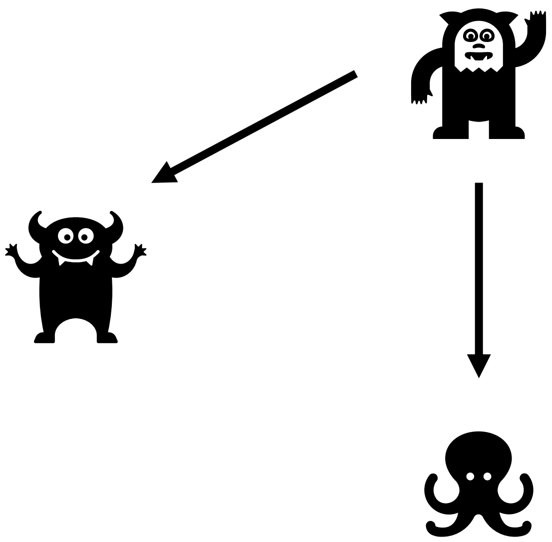
Example

						…
	3	15	**4**	**5**	8	…
	**2**	11	5	3	**5**	…
	6	**11**	1	4	7	…



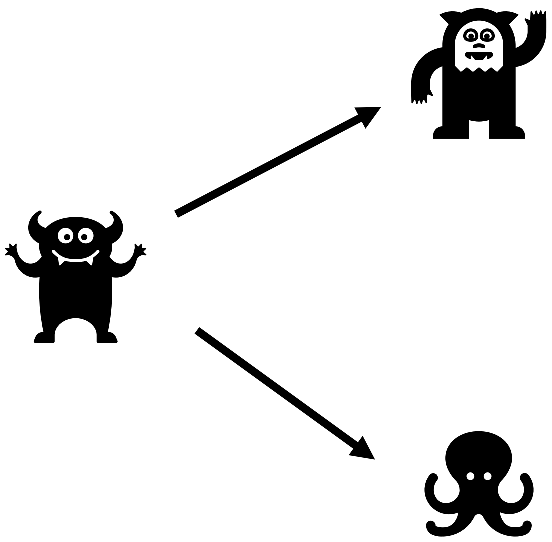
Example

	3	15	4	5	8	…
	2	11	5	3	5	…
	6	11	1	4	7	…



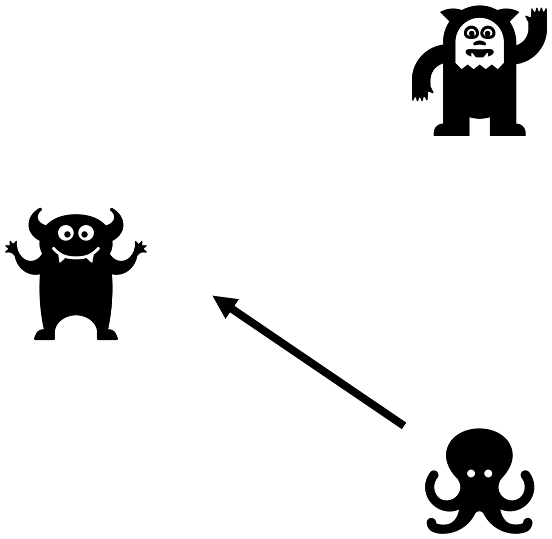
Example

						…
	3	15	4	5	8	…
	2	11	5	3	5	…
	6	11	1	4	7	…



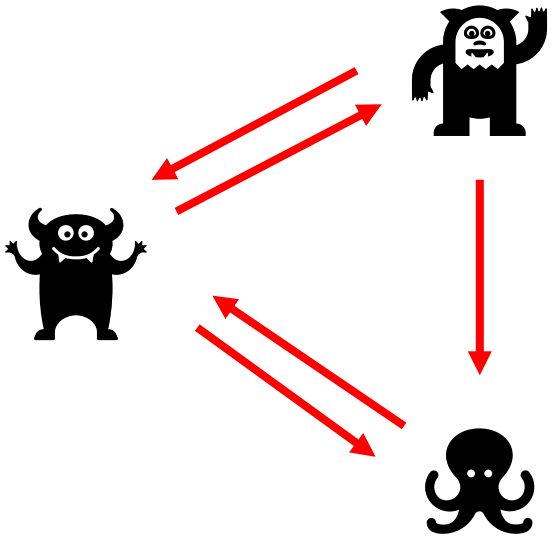
Example

	3	15	**4**	**5**	8	…
	2	11	5	3	**5**	…
	6	**11**	1	4	7	…



Example

	3	15	**4**	**5**	8	…
	2	11	5	3	**5**	…
	6	**11**	1	4	7	…



Computation of EFX Allocations

Envy-Cycle Elimination [LiptonMMS, EC 2004]

Envy-graph for a given (partial) allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$

- Directed graph $G(N, E)$; $(i, j) \in E$ if i envies j $(v_i(X_i) < v_i(X_j))$
- Sink (in-degree = 0) node: not envied by any other agent
- No sink ⇒ exist a cycle

Computation of EFX Allocations

Envy-Cycle Elimination [LiptonMMS, EC 2004]

Envy-graph for a given (partial) allocation $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$

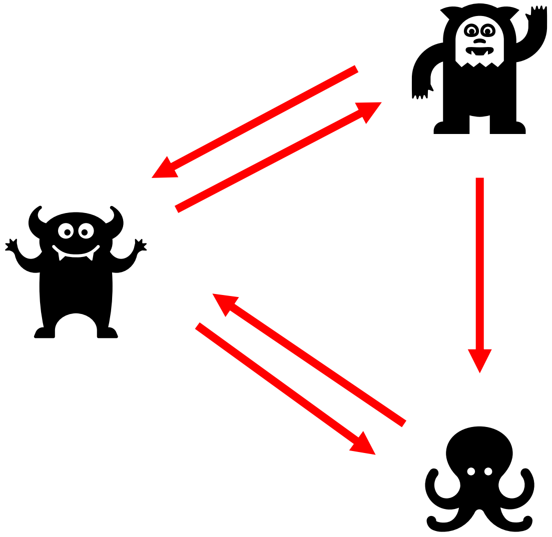
- Directed graph $G(N, E)$; $(i, j) \in E$ if i envies j $(v_i(X_i) < v_i(X_j))$
- Sink (in-degree = 0) node: not envied by any other agent
- No sink ⇒ exists a cycle

Envy-cycle elimination: if $(i, j) \in E$ is in the cycle, let agent i get bundle X_j

"Everyone in the cycle gets what she wants"

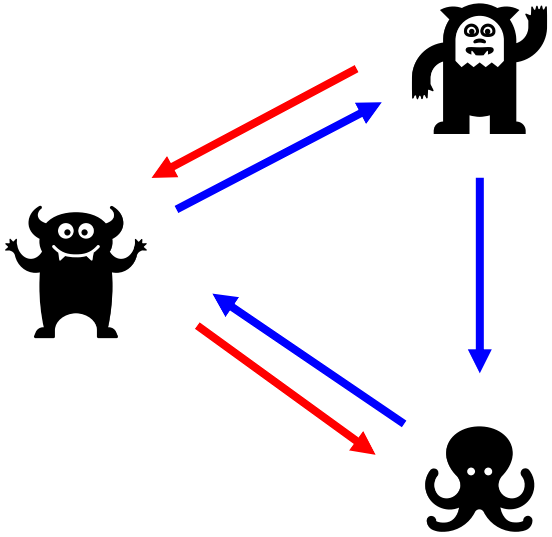
Example

						…
	3	15	**4**	**5**	8	…
	2	11	5	3	**5**	…
	6	**11**	1	4	7	…



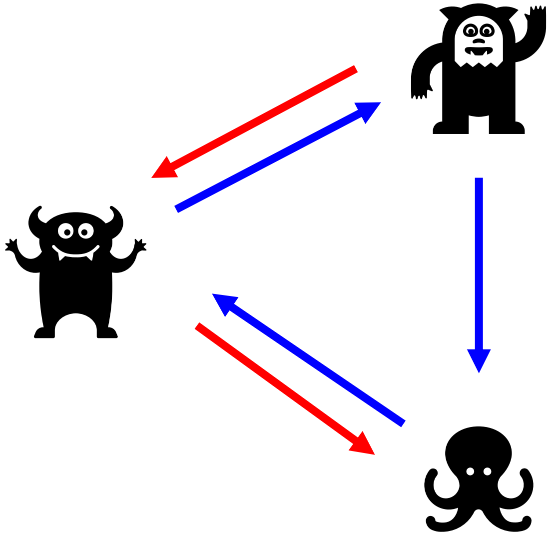
Example

	3	15	**4**	**5**	8	…
	2	11	5	3	**5**	…
	6	**11**	1	4	7	…



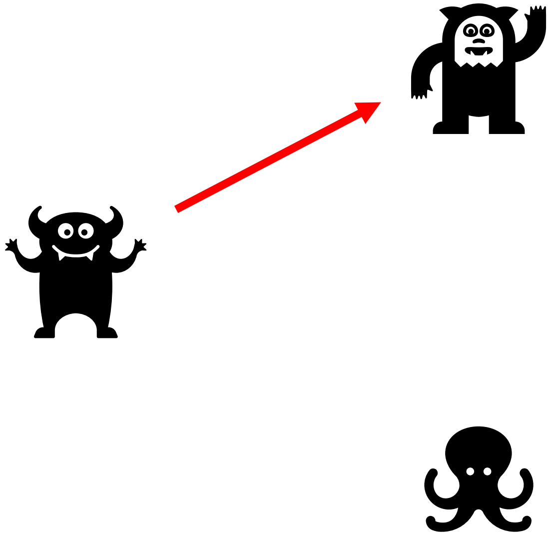
Example

						…
	3	15	4	5	8	…
	2	11	5	3	5	…
	6	11	1	4	7	…



Example

						…
	3	**15**	4	5	8	…
	2	11	**5**	**3**	5	…
	6	11	1	4	**7**	…



Computation of EFX Allocations

Envy-Cycle Elimination [LiptonMMS, EC 2004]

Envy-graph for a given (partial) allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$

- Directed graph $G(N, E)$; $(i, j) \in E$ if i envies j $(v_i(X_i) < v_i(X_j))$
- Sink (in-degree = 0) node: not envied by any other agent
- No sink ⇒ exists a cycle

Envy-cycle elimination: if $(i, j) \in E$ is in the cycle, let agent i get bundle X_j

- $|E|$ decreases by at least one after each elimination
- Repeat until the graph becomes acyclic (contains a sink node)

Computation of EFX Allocations

Identical Ordering (IDO) instances:

- for all agent $i \in N$: $v_i(e_1) \geq v_i(e_2) \geq \cdots \geq v_i(e_m)$.

Initialize $X_i \leftarrow \emptyset$ for all $i \in N$

for $t = 1,2, \ldots, m$:

- construct the envy-graph on $\boldsymbol{X}$
- use envy-cycle elimination to remove cycles and find a sink node i
- update $X_i \leftarrow X_i \cup \{e_t\}$

The resulting allocation is EFX because

Consider any agent $j \in N$ with bundle X_j

- Let $e_t \in M$ be the last item agent j receives
- For all $i \neq j$, we have $v_i(X_i) \geq v_i\left(X_j \setminus \{e_t\}\right)$ (because j was sink)
- For all $e \in X_i$, $v_j(e) \geq v_j(e_t)$ (for IDO instances)

[**EFX**] For all agents $i, j \in N$ and $e \in X_j$, $v_i(X_i) \geq v_i\left(X_j \setminus \{e\}\right)$

Extensions of Envy-Cycle Elimination

Envy-Cycle Elimination [LiptonMMS, EC 2004]

Champion Graphs [ChaudhuryGM, EC 2020; BergerCFF, AAAI 2022]

Rainbow Cycle Number [ChaudhuryGMMM, EC 2021]

Top-Trading Envy-Cycle Elimination [BhaskarSV, APPROX 2021]

Other variants [BarmanBMN, AAAI 2018; AmanatidisMN, TCS 2020]

Existence of EFX Allocations

EFX allocation for IDO instances and for two agents [Plaut and Roughgarden, SIDMA 2020]

EFX allocation for three agents [ChaudhuryGM, EC 2020]

EFX allocation for bi-valued instances [AmanatidisBFHV, TCS 2021]

Existence of EFX Allocations

EFX allocation for IDO instances and for two agents [Plaut and Roughgarden, SIDMA 2020]

EFX allocation for three agents [ChaudhuryGM, EC 2020]

EFX allocation for bi-valued instances [AmanatidisBFHV, TCS 2021]

OPEN Do EFX allocations always exist ?

Existence of EFX Allocations

EFX allocation for IDO instances and for two agents [Plaut and Roughgarden, SIDMA 2020]

EFX allocation for three agents [ChaudhuryGM, EC 2020]

EFX allocation for bi-valued instances [AmanatidisBFHV, TCS 2021]

Approximations: α-EFX allocations, for $\alpha \in (0,1)$

Partial allocations: EFX allocations that leave some items unallocated

Approximation of EFX Allocations

For $\alpha \in (0,1)$, α-Approximate Envy-free up to any item (**α-EFX**) : $\forall i \in N$, $\forall j \in N$:

$$\forall e \in X_j : v_i(X_i) \geq \alpha \cdot v_i\left(X_j \setminus \{e\}\right)$$

0.5-EFX [Plaut and Roughgarden, SIDMA 2020; ChanCLW, IJCAI 2019]

0.618-EFX [AmanatidisMN, TCS 2020]

Approximation of EFX Allocations

For $\alpha \in (0,1)$, α-Approximate Envy-free up to any item (**α-EFX**) : $\forall i \in N$, $\forall j \in N$:

$$\forall e \in X_j : v_i(X_i) \geq \alpha \cdot v_i\left(X_j \setminus \{e\}\right)$$

0.5-EFX [Plaut and Roughgarden, SIDMA 2020; ChanCLW, IJCAI 2019]

0.618-EFX [AmanatidisMN, TCS 2020]

OPEN What is the best approximation ratio for EFX?

Partial EFX Allocations

Partial allocation $\boldsymbol{X} = (X_1, \dots, X_n)$: $P = M \setminus (\cup_i X_i) \neq \emptyset$

- P contains the unallocated items / items donated to the **charity**
- **High quality allocation**: e.g., large (Nash) social welfare of X, small P

Partial EFX Allocations

Partial allocation $\boldsymbol{X} = (X_1, \dots, X_n)$: $P = M \setminus (\cup_i X_i) \neq \emptyset$

- P contains the unallocated items / items donated to the **charity**
- **High quality allocation**: e.g., large (Nash) social welfare of X, small P

EFX partial allocation with half Max-NSW [CaragiannisGH, EC 2019]

EFX partial allocation with $|P| \leq n - 1$ [ChaudhuryKMS, SICOMP 2021]

$(1 - \epsilon)$-EFX partial allocation with $|P| = o(n)$ [ChaudhuryGMMM, EC 2021]

EFX partial allocation for 4 agents with $|P| = 1$ [BergerCFF, AAAI 2022]

Existence of EFX Allocations for Chores

EFX Allocations Always Exist for

- IDO valuations [LiLW, WWW 2022]
- 2 agents (with general valuations) [Plaut and Roughgarden, SIDMA 2020]
- 3 agents with bi-valued valuation functions [Zhou and Wu, IJCAI 2022]
- Levelled instances [GafniHLT, 2021]

Existence of EFX Allocations for Chores

EFX Allocations Always Exist for

- IDO valuations [LiLW, WWW 2022]
- 2 agents (with general valuations) [Plaut and Roughgarden, SIDMA 2020]
- 3 agents with bi-valued valuation functions [Zhou and Wu, IJCAI 2022]
- Levelled instances [GafniHLT, 2021]

Approximations:

- 5-EFX for 3 agents; $O(n^2)$-EFX for n agents [Zhou and Wu, IJCAI 2022]

Existence of EFX Allocations for Chores

EFX Allocations Always Exist for

- IDO valuations [LiLW, WWW 2022]
- 2 agents (with general valuations) [Plaut and Roughgarden, SIDMA 2020]
- 3 agents with bi-valued valuation functions [Zhou and Wu, IJCAI 2022]
- Levelled instances [GafniHLT, 2021]

Approximations:

- 5-EFX for 3 agents; $O(n^2)$-EFX for n agents [Zhou and Wu, IJCAI 2022]

OPEN Do EFX allocations always exist for chores?

Existence of EFX Allocations for Chores

EFX Allocations Always Exist for

- IDO valuations [LiLW, WWW 2022]
- 2 agents (with general valuations) [Plaut and Roughgarden, SIDMA 2020]
- 3 agents with bi-valued valuation functions [Zhou and Wu, IJCAI 2022]
- Levelled instances [GafniHLT, 2021]

Approximations:

- 5-EFX for 3 agents; $O(n^2)$-EFX for n agents [Zhou and Wu, IJCAI 2022]

OPEN Do $O(1)$-EFX allocations always exist for chores?

Approximation of MMS Allocations

Maximin Share (MMS) of agent $i \in N$

Let $\Pi_n(M)$ be the set of all n-partitions of items in M:

$$\mathrm{MMS}_i(M,n) = \max_{(X_1,\ldots,X_n)\in\Pi_n(M)} \min_{j\in N}\{v_i(X_j)\}$$

For $\alpha \in (0,1]$, allocation $\boldsymbol{X}$ is **$\boldsymbol{\alpha}$-MMS** if $v_i(X_i) \geq \alpha \cdot \mathrm{MMS}_i$ for all $i \in N$

Approximation of MMS Allocations

Theorem [Reduction to IDO instance]:

Algorithm that computes an α-MMS allocation for every IDO instance

⇒ Algorithm that computes an α-MMS allocation for general instances

Approximation of MMS Allocations

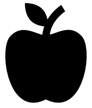
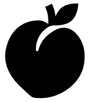
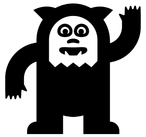
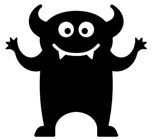
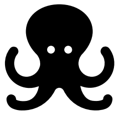
Theorem [Reduction to IDO instance]:

Algorithm that computes an α-MMS allocation for every IDO instance

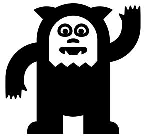
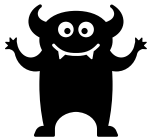
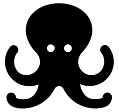
⇒ Algorithm that computes an α-MMS allocation for general instances

- Given any general instance, construct an IDO instance: $v_i(e_1) \geq v_i(e_2) \geq \cdots \geq v_i(e_m)$
- Compute an α-MMS allocation $\boldsymbol{X}$ (on the IDO instance)
- For $j = 1,2, \ldots, m$, set $\sigma(j) \leftarrow i$ if $e_j \in X_i$
- Run the sequential picking algorithm with σ on the original instance

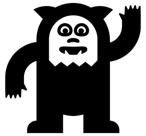
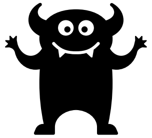
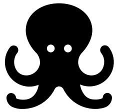
Example: Original Instance

							**MMS**
	3	15	4	5	8	10	**15**
	2	11	5	3	5	7	**10**
	6	11	1	4	7	9	**12**

Example: IDO Instance

	e_1	e_2	e_3	e_4	e_5	e_6	**MMS**
	15	10	8	5	4	3	**15**
	11	7	5	5	3	2	**10**
	11	9	7	6	4	1	**12**

Example: Allocation for IDO Instance

	e_1	e_2	e_3	e_4	e_5	e_6	**MMS**
	**15**	10	8	5	4	3	**15**
	11	**7**	5	**5**	3	2	**10**
	11	9	**7**	6	**4**	**1**	**12**

Example: Picking Sequence

For the IDO instance: $X_1 = \{e_1\}$, $X_2 = \{e_2, e_4\}$, $X_3 = \{e_3, e_5, e_6\}$

In the picking sequence:

- $\sigma(1) = 1, \sigma(2) = \sigma(4) = 2, \sigma(3) = \sigma(5) = \sigma(6) = 3$

Example: Picking Sequence

For the IDO instance: $X_1 = \{e_1\}$, $X_2 = \{e_2, e_4\}$, $X_3 = \{e_3, e_5, e_6\}$

In the picking sequence:

- $\sigma(1) = 1, \sigma(2) = \sigma(4) = 2, \sigma(3) = \sigma(5) = \sigma(6) = 3$

Agent 1 gets to pick an item in Round-1

Agent 2 gets to pick an item in Round-2 and Round-4

Agent 3 gets to pick an item in Round-3, Round-5 and Round-6

Example: Picking Sequence

For the IDO instance: $X_1 = \{e_1\}, X_2 = \{e_2, e_4\}, X_3 = \{e_3, e_5, e_6\}$

In the picking sequence:

- $\sigma(1) = 1, \sigma(2) = \sigma(4) = 2, \sigma(3) = \sigma(5) = \sigma(6) = 3$

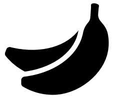
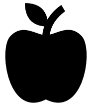
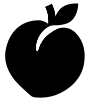
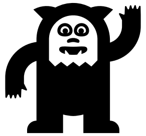
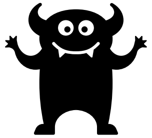
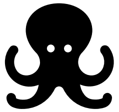
Agent 1 gets to pick an item in Round-1

Agent 2 gets to pick an item in Round-2 and Round-4

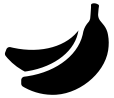
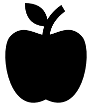
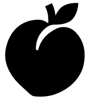
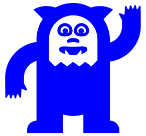
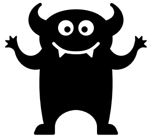
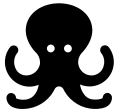
Agent 3 gets to pick an item in Round-3, Round-5 and Round-6

The item an agent i picks in Round-j is at least as good as $v_i(e_j)$

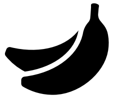
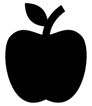
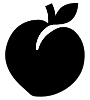
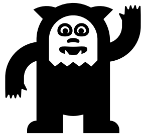
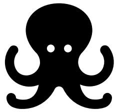
Example: Original Instance

							**MMS**
	3	15	4	5	8	10	**15**
	2	11	5	3	5	7	**10**
	6	11	1	4	7	9	**12**

Example: Original Instance $\sigma(1) = 1$

							MMS
	3	15	4	5	8	10	15
	2	11	5	3	5	7	10
	6	11	1	4	7	9	12

Example: Original Instance $\sigma(2) = 2$

							MMS
	3	15	4	5	8	10	15
	2	11	5	3	5	7	10
	6	11	1	4	7	9	12

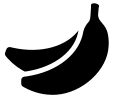
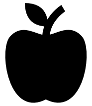
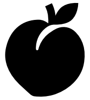
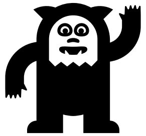
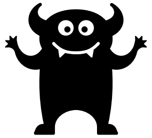
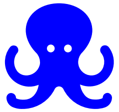
Example: Original Instance $\sigma(3) = 3$

							MMS
	3	**15**	4	5	8	10	**15**
	2	11	5	3	5	**7**	**10**
	6	11	1	4	**7**	9	**12**

Example: Original Instance $\sigma(4) = 2$

							MMS
	3	**15**	4	5	8	10	**15**
	2	11	**5**	3	5	**7**	**10**
	6	11	1	4	**7**	9	**12**

Example: Original Instance $\sigma(5) = \sigma(6) = 3$

							**MMS**
	3	**15**	4	5	8	10	**15**
	2	11	**5**	3	5	**7**	**10**
	**6**	11	1	**4**	**7**	9	**12**

Approximation of MMS Allocations

MMS Allocation is not guaranteed to exist [KurokawaPW, JACM 2018]

2/3-MMS [KurokawaPW, JACM 2018; GargMT, SOSA 2019]

3/4-MMS [GhodsiHSSY, EC 2018]

(3/4+1/(12n))-MMS [Garg and Taki, AIJ 2021]

Upper bound on approximation ratio: 39/40 [FeigeST, WINE 2021]

Approximation of MMS Allocations (Chores)

MMS Allocation is not guaranteed to exist [AzizRSW, AAAI 2017]

2-MMS allocation from PROP1/EF1 allocation

$4/3$-MMS allocation computation [Barman and Murthy, EC 2017]

$11/9$-MMS allocation exists [Huang and Lu, EC 2021]

Lower bound on approximation ratio: 44/43 [FeigeST, WINE 2021]

Other Settings & Extensions

Advanced Settings

Fair and Efficient Allocations

Weighted/Asymmetric Agents

Budget-Feasible Setting

Ordinal Preference Settings

…

Fair and Efficient Allocations

Efficiency Measurements

For allocation allocation $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$

- Social Welfare: $\mathrm{SW}(\boldsymbol{X}) = \sum_{i \in N} v_i(X_i)$
- Nash Social Welfare: $\mathrm{NSW}(\boldsymbol{X}) = \Pi_{i \in N} v_i(X_i)$

Allocation $\boldsymbol{Y} = (Y_1, Y_2, \dots, Y_n)$ dominates $\boldsymbol{X}$

- if $v_i(Y_i) \geq v_i(X_i)$ for all i and $v_i(Y_i) > v_i(X_i)$ for some i

An allocation is Pareto optimal (PO) if it is not dominated by any allocation.

Fair and Efficient Allocations

Efficiency guarantee of the allocation (in addition to being fair)?

E.g., EF1 allocations with high (Nash) social welfare or PO guarantees.

Fair and Efficient Allocations

Efficiency guarantee of the allocation (in addition to being fair)?

E.g., EF1 allocations with high (Nash) social welfare or PO guarantees.

The Max-NSW allocation is EF1 & PO [CaragiannisMPSW, TAEC 2019].

Fair and Efficient Allocations

Proof. PO: the Max-NSW allocation is not dominated by any other allocation

Fair and Efficient Allocations

Proof. **PO**: the Max-NSW allocation is not dominated by any other allocation

EF1: suppose the allocation is not EF1: $\forall e \in X_j,\ v_i(X_i) < v_i\left(X_j \setminus \{e\}\right)$.

- Let $e^* = \arg\max_{e \in X_j} \left\{ \frac{v_i(e)}{v_j(e)} \right\}$

Fair and Efficient Allocations

Proof. **PO**: the Max-NSW allocation is not dominated by any other allocation

EF1: suppose the allocation is not EF1: $\forall e \in X_j,\ v_i(X_i) < v_i(X_j \setminus \{e\})$.

- Let $e^* = \arg\max_{e \in X_j} \left\{ \frac{v_i(e)}{v_j(e)} \right\}$
- $X_i' \leftarrow X_i \cup \{e^*\},\ X_j' \leftarrow X_j \setminus \{e^*\}$

Need to show: $v_i(X_i') \cdot v_j(X_j') > v_i(X_i) \cdot v_j(X_j)$

Fair and Efficient Allocations

Analysis. $(X_i' \leftarrow X_i \cup \{e^*\}, X_j' \leftarrow X_j \setminus \{e^*\})$

- Let $e^* = \arg\max_{e \in X_j} \left\{\frac{v_i(e)}{v_j(e)}\right\}$ $\Rightarrow$ $\frac{v_i(e^*)}{v_j(e^*)} \geq \frac{v_i(X_j)}{v_j(X_j)}$

Fair and Efficient Allocations

Analysis. $(X_i' \leftarrow X_i \cup \{e^*\}, X_j' \leftarrow X_j \setminus \{e^*\})$

- Let $e^* = \arg\max_{e \in X_j} \left\{ \frac{v_i(e)}{v_j(e)} \right\} \Rightarrow \quad \frac{v_i(e^*)}{v_j(e^*)} \geq \frac{v_i(X_j)}{v_j(X_j)} \quad \Rightarrow \quad \frac{v_i(e^*)}{v_i(X_j)} \geq \frac{v_j(e^*)}{v_j(X_j)}$

Fair and Efficient Allocations

Analysis. $(X_i' \leftarrow X_i \cup \{e^*\}, X_j' \leftarrow X_j \setminus \{e^*\})$

- Let $e^* = \arg\max_{e \in X_j} \left\{ \frac{v_i(e)}{v_j(e)} \right\}$ $\Rightarrow$ $\frac{v_i(e^*)}{v_j(e^*)} \geq \frac{v_i(X_j)}{v_j(X_j)}$ $\Rightarrow$ $\frac{v_i(e^*)}{v_i(X_j)} \geq \frac{v_j(e^*)}{v_j(X_j)}$
- Let $a = v_i(e^*)$ and $b = v_i(X_i)$ $\Rightarrow$ $v_i(X_j) > a + b$ (by non-EF1)

Fair and Efficient Allocations

Analysis. $(X_i' \leftarrow X_i \cup \{e^*\}, X_j' \leftarrow X_j \setminus \{e^*\})$

- Let $e^* = \arg\max_{e \in X_j} \left\{\frac{v_i(e)}{v_j(e)}\right\} \Rightarrow \frac{v_i(e^*)}{v_j(e^*)} \geq \frac{v_i(X_j)}{v_j(X_j)} \Rightarrow \frac{v_i(e^*)}{v_i(X_j)} \geq \frac{v_j(e^*)}{v_j(X_j)}$
- Let $a = v_i(e^*)$ and $b = v_i(X_i) \Rightarrow v_i(X_j) > a + b$ (by non-EF1)
- $v_i(X_i') = \left(1 + \frac{a}{b}\right) \cdot v_i(X_i) = \left(\frac{a+b}{b}\right) \cdot v_i(X_i)$

Fair and Efficient Allocations

Analysis. $(X_i' \leftarrow X_i \cup \{e^*\}, X_j' \leftarrow X_j \setminus \{e^*\})$

- Let $e^* = \arg\max_{e \in X_j} \left\{\frac{v_i(e)}{v_j(e)}\right\}$ $\Rightarrow$ $\frac{v_i(e^*)}{v_j(e^*)} \geq \frac{v_i(X_j)}{v_j(X_j)}$ $\Rightarrow$ $\frac{v_i(e^*)}{v_i(X_j)} \geq \frac{v_j(e^*)}{v_j(X_j)}$
- Let $a = v_i(e^*)$ and $b = v_i(X_i)$ $\Rightarrow$ $v_i(X_j) > a + b$ (by non-EF1)
- $v_i(X_i') = \left(1 + \frac{a}{b}\right) \cdot v_i(X_i) = \left(\frac{a+b}{b}\right) \cdot v_i(X_i)$
- $v_j(X_j') = \left(1 - \frac{v_j(e^*)}{v_j(X_j)}\right) \cdot v_j(X_j) > \left(1 - \frac{a}{a+b}\right) \cdot v_j(X_j) = \left(\frac{b}{a+b}\right) \cdot v_j(X_j)$

Fair and Efficient Allocations

Efficiency guarantee of the allocation (in addition to being fair)?

E.g., EF1 allocations with high (Nash) social welfare or PO guarantees.

The Max-NSW allocation is EF1 & PO [CaragiannisMPSW, TAEC 2019].

Pseudo-polynomial time algorithm for computation of EF1 & PO [BarmanKV, EC 2018]

PROP1 & PO for mixture of goods and chores [AzizMS, ORL 2020]

EF1 & PO for bi-valued chores [EbadianPS, AAMAS 2022; GargMQ, AAAI 2022]

Fair and Efficient Allocations

Efficiency guarantee of the allocation (in addition to being fair)?

E.g., EF1 allocations with high (Nash) social welfare or PO guarantees.

The Max-NSW allocation is EF1 & PO [CaragiannisMPSW, TAEC 2019].

Pseudo-polynomial time algorithm for computation of EF1 & PO [BarmanKV, EC 2018]

PROP1 & PO for mixture of goods and chores [AzizMS, ORL 2020]

EF1 & PO for bi-valued chores [EbadianPS, AAMAS 2022; GargMQ, AAAI 2022]

OPEN Polynomial-time algorithms for computing EF1 & PO ?

Fair and Efficient Allocations

Efficiency guarantee of the allocation (in addition to being fair)?
E.g., EF1 allocations with high (Nash) social welfare or PO guarantees.

The Max-NSW allocation is EF1 & PO [CaragiannisMPSW, TAEC 2019].
Pseudo-polynomial time algorithm for computation of EF1 & PO [BarmanKV, EC 2018]
PROP1 & PO for mixture of goods and chores [AzizMS, ORL 2020]
EF1 & PO for bi-valued chores [EbadianPS, AAMAS 2022; GargMQ, AAAI 2022]

OPEN Existence of EF1 & PO allocations for chores?

Fair and Efficient Allocations

Efficiency guarantee of the allocation (in addition to being fair)?

E.g., EF1 allocations with high (Nash) social welfare or PO guarantees.

The Max-NSW allocation is EF1 & PO [CaragiannisMPSW, TAEC 2019].

Pseudo-polynomial time algorithm for computation of EF1 & PO [BarmanKV, EC 2018]

PROP1 & PO for mixture of goods and chores [AzizMS, ORL 2020]

EF1 & PO for bi-valued chores [EbadianPS, AAMAS 2022; GargMQ, AAAI 2022]

OPEN Existence of PROPX & PO allocations for chores?

Weighted (Asymmetric) Agents

Agents with Asymmetric Weights

Each agent $i \in N$ has a weight $s_i > 0$ and $\sum_i s_i = 1$

- Unweighted case: $s_i = 1/n$ for all $i \in N$

Weighted PROP: $v_i(X_i) \geq s_i \cdot v_i(M)$ for all $i \in N$

- Extends naturally to PROP1 and PROPX

Agents with Asymmetric Weights

Each agent $i \in N$ has a weight $s_i > 0$ and $\sum_i s_i = 1$

- Unweighted case: $s_i = 1/n$ for all $i \in N$

Weighted PROP: $v_i(X_i) \geq s_i \cdot v_i(M)$ for all $i \in N$

- Extends naturally to PROP1 and PROPX

Weighted EF: for all $i, j \in N$, $\frac{v_i(X_i)}{s_i} \geq \frac{v_i(X_j)}{s_j}$

- Extends naturally to EF1 and EFX

Agents with Asymmetric Weights

Each agent $i \in N$ has a weight $s_i > 0$ and $\sum_i s_i = 1$

- Unweighted case: $s_i = 1/n$ for all $i \in N$

Weighted PROP: $v_i(X_i) \geq s_i \cdot v_i(M)$ for all $i \in N$

- Extends naturally to PROP1 and PROPX

Weighted EF: for all $i, j \in N$, $\frac{v_i(X_i)}{s_i} \geq \frac{v_i(X_j)}{s_j}$

- Extends naturally to EF1 and EFX

Other notions: WMMS [FarhadiGHLPSSY, JAIR 2019], APS [BabaioffEF, EC 2021]

Agents with Asymmetric Weights

Computation of allocations that are

- WEF1 [ChakrabortyISZ, TEAC 2021]
- WPROP1 for mixture of goods and chores [AzizMS, ORL 2020]
- WPROPX for chores [LiLW, WWW 2022]

Agents with Asymmetric Weights

Computation of allocations that are

- WEF1 [ChakrabortyISZ, TEAC 2021]
- WPROP1 for mixture of goods and chores [AzizMS, ORL 2020]
- WPROPX for chores [LiLW, WWW 2022]

OPEN Best possible approximations for weighted fairness notions?

Budget-Feasible Setting

Budget-Feasible Allocations

Each item $e \in M$ has a **size** s_e; each agent i has a **capacity** C_i

The total size of items in X_i should not exceed the capacity of agent i

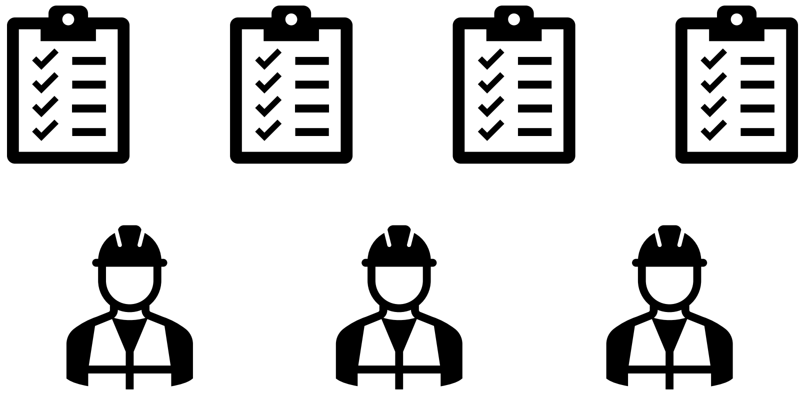
Budget-Feasible Allocations

Each item $e \in M$ has a **size** s_e; each agent i has a **capacity** C_i

The total size of items in X_i should not exceed the capacity of agent i

Applications:

- Items → tasks
  - Value = payment; size = processing time
- Agents → workers
  - Capacity = capability



Budget-Feasible Allocations

Under capacity constraints :

- Agent i envies agent j if $T \subseteq X_j$ with $s(T) = \sum_{e \in T} s_e \leq C_i$, such that $v_i(X_i) < v_i(T)$
- Some items are unallocated (donated to the charity)
- EF1 allocation: no agents envies another agent or charity by more than one item

Budget-Feasible Allocations

Under capacity constraints :

- Agent i envies agent j if $T \subseteq X_j$ with $s(T) = \sum_{e \in T} s_e \leq C_i$, such that $v_i(X_i) < v_i(T)$
- Some items are unallocated (donated to the charity)
- EF1 allocation: no agents envies another agent or charity by more than one item

Max-NSW allocation is 1/4-EF1 and PO [WuLG, IJCAI 2021]

Computation of 1/2-EF1 allocation for identical valuations [GanLW, 2021]

Budget-Feasible Allocations

Under capacity constraints :

- Agent i envies agent j if $T \subseteq X_j$ with $s(T) = \sum_{e \in T} s_e \leq C_i$, such that $v_i(X_i) < v_i(T)$
- Some items are unallocated (donated to the charity)
- EF1 allocation: no agents envies another agent or charity by more than one item

Max-NSW allocation is 1/4-EF1 and PO [WuLG, IJCAI 2021]

Computation of 1/2-EF1 allocation for identical valuations [GanLW, 2021]

OPEN EF1 allocation for identical valuations ?

Budget-Feasible Allocations

Under capacity constraints :

- Agent i envies agent j if $T \subseteq X_j$ with $s(T) = \sum_{e \in T} s_e \leq C_i$, such that $v_i(X_i) < v_i(T)$
- Some items are unallocated (donated to the charity)
- EF1 allocation: no agents envies another agent or charity by more than one item

Max-NSW allocation is 1/4-EF1 and PO [WuLG, IJCAI 2021]

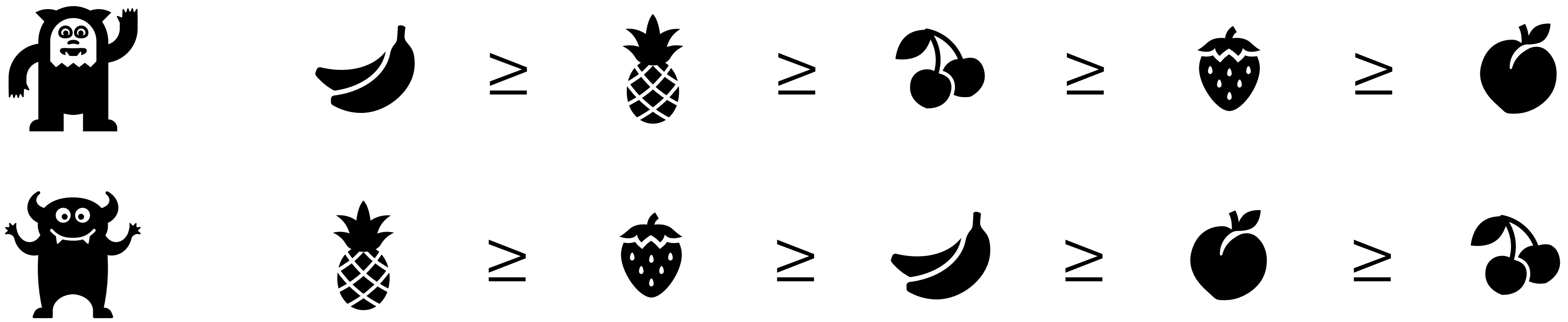
Computation of 1/2-EF1 allocation for identical valuations [GanLW, 2021]

OPEN EF1 allocation under general valuations ?

Ordinal Preference Settings

Ordinal Approximation Algorithm

How to compute a fair allocation with only **ordinal information** ?



Ordinal Approximation Algorithm

Given only the ordinal preferences of agents

Compute an α-MMS allocation

- For any valuations that agree with the ordinal preferences, the allocation is α-MMS
- MMS value of each agent is defined by the cardinal values

Ordinal Algorithms for Approximating MMS for chores [AzizLW, 2020]

Ordinal Algorithm for α-MMS for Chores

W.l.o.g., we only need to consider Identical Ordinal (IDO) Preference:

$$\forall i \in N, c_i(e_1) \geq c_i(e_2) \geq \cdots \geq c_i(e_m)$$

Round-Robin is $\left(2 - \frac{1}{n}\right)$-approximate

- Allocation sequence: $\underbrace{(1,2, \dots, n, 1,2, \dots, n, \dots)}_{\text{length } m}$

Ordinal Algorithm for α-MMS for Chores

W.l.o.g., we only need to consider Identical Ordinal (IDO) Preference:

$$\forall i \in N, c_i(e_1) \geq c_i(e_2) \geq \cdots \geq c_i(e_m)$$

Round-Robin is $\left(2 - \frac{1}{n}\right)$-approximate

- Allocation sequence: $(1,2, \ldots, n, 1,2, \ldots, n, \ldots) = (1,2, \ldots, n)^*$
- $(1,2, \ldots, n)$ is the **pattern** of the sequence

Ordinal Algorithm for α-MMS for Chores

[AzizLW, 2020] There exists a pattern (depends on n) for which the allocation sequence is

- 1.33-MMS for $n = 2$ (optimal)
- 1.4-MMS for $n = 3$ (optimal)
- 1.5-MMS for $n = 4$ (lower bound: 1.405)
- 1.66-MMS for $n \geq 5$

Ordinal Algorithm for α-MMS for Chores

[AzizLW, 2020] There exists a pattern (depends on n) for which the allocation sequence is

- 1.33-MMS for $n = 2$ (optimal)
- 1.4-MMS for $n = 3$ (optimal)
- 1.5-MMS for $n = 4$ (lower bound: 1.405)
- 1.66-MMS for $n \geq 5$

OPEN Optimal Ratios for $n \geq 4$ agents?

Ordinal Algorithm for α-MMS for Chores

[AzizLW, 2020] There exists a pattern (depends on n) for which the allocation sequence is

- 1.33-MMS for $n = 2$ (optimal)
- 1.4-MMS for $n = 3$ (optimal)
- 1.5-MMS for $n = 4$ (lower bound: 1.405)
- 1.66-MMS for $n \geq 5$

OPEN Ordinal Approximations of other fairness notions?

Thank You

Questions?