Existence and Computation of Epistemic EFX Allocations

Ioannis Caragiannis
Aarhus University

Nidhi Rathi
MPI-INF

Giovanna Varricchio
Goethe University Frankfurt

Jugal Garg
University of Illinois

Eklavya Sharma
University of Illinois

by Naya Rudolph (she/her)

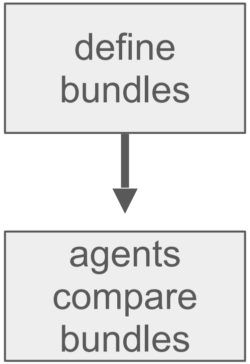
Presentation outline

- Background information and setting
- Issues with EF, EFX, EF1
- Epistemic EFX
  - the idea
  - definition
  - algorithm
  - analysis
- MMS implies EEFX
- Take home messages

Fairness has two main approaches

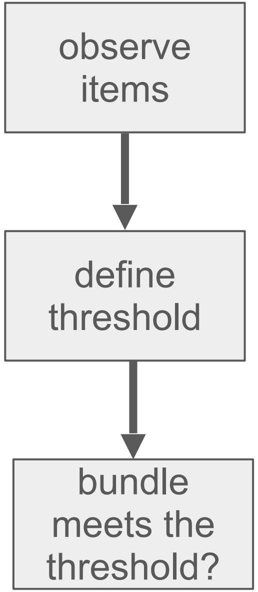
<u>Comparative</u>

EF, EF1, EFX, EEFX(!)



<u>In absolute terms</u>

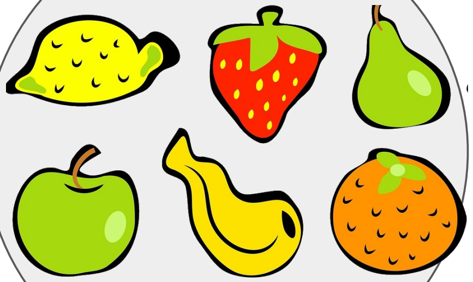
PROP, MMS, PROP1, MXS(!)



Our setting is indivisible goods

<u>non-zero</u> positive valuations

$\forall i \in [n], g \in [m]$:
$v_i(g) > 0$

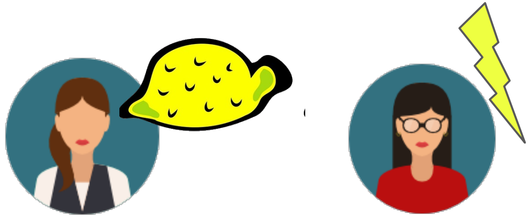


additive valuations

$\forall i \in [n], S \subseteq [m]$:
$v_i(S) = \sum_{s \in S} v_i(s)$

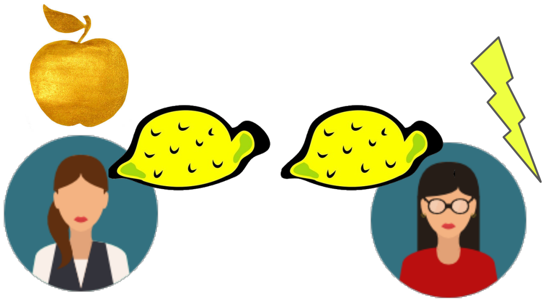
Relaxations of EF have their problems

EF does not always exist



- ➢ NP hard (partition problem)

EF1 is not that fair



EFX might not always exist

- ➢ NP hard
- ➢ exists for 3 agents

A new relaxation by removing epistemic access

What knowledge does the agent have:

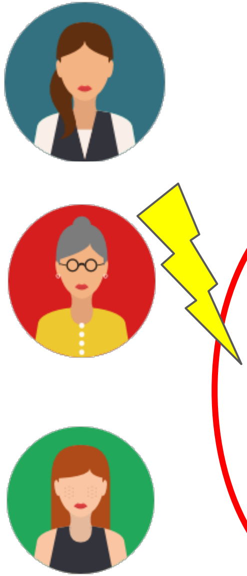
- agents own bundle
- amount of agents
- all items
- ~~bundles of other agents~~

➢ The agent does not know what the other bundles look like anymore

Excursion:

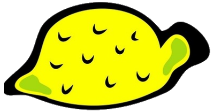
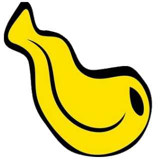
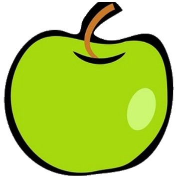
- epistemic:
  relating to knowledge
- epistemic access:
  access to knowledge about a situation or reality

Is this allocation EFX?

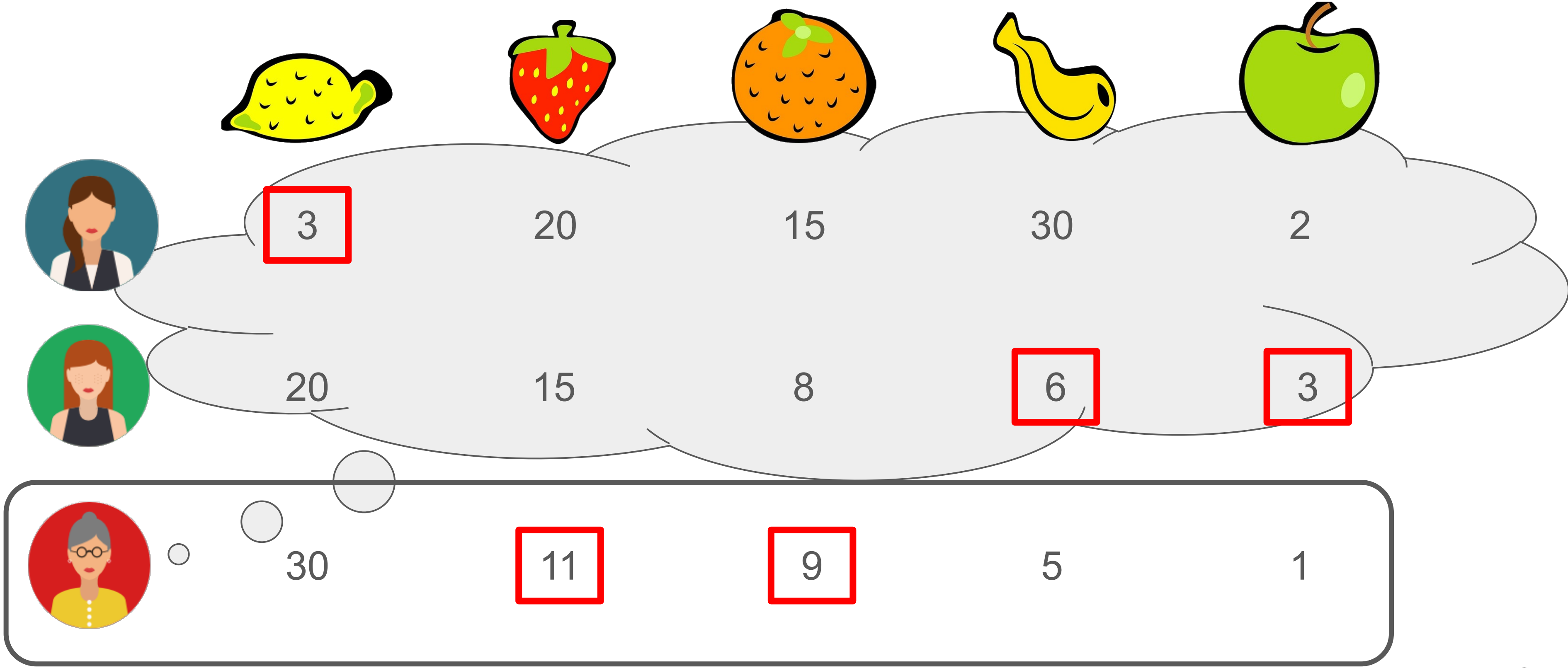


3	20	15	30	2
30	11	9	5	1
20	15	8	6	3

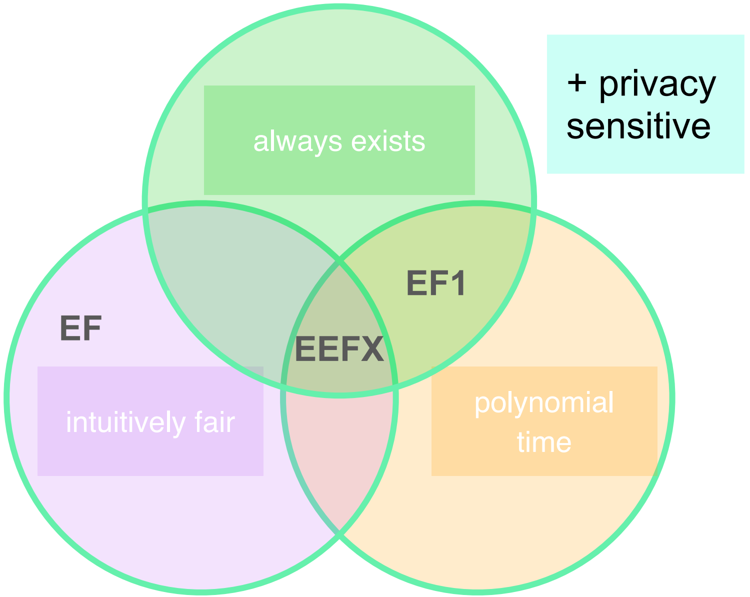
Remove Red’s epistemic access

					
	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

Remove Red’s epistemic access

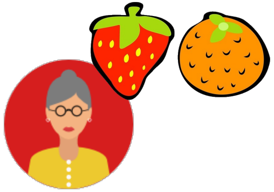


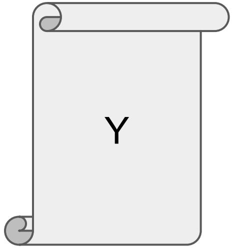
EEFX is fair and can be compute in polynomial time



X needs an EEFX certificate for each agent to be EEFX

- set of agents [n]
- set of goods [m]
- allocation $X = (X_1, X_2, \ldots, X_n)$





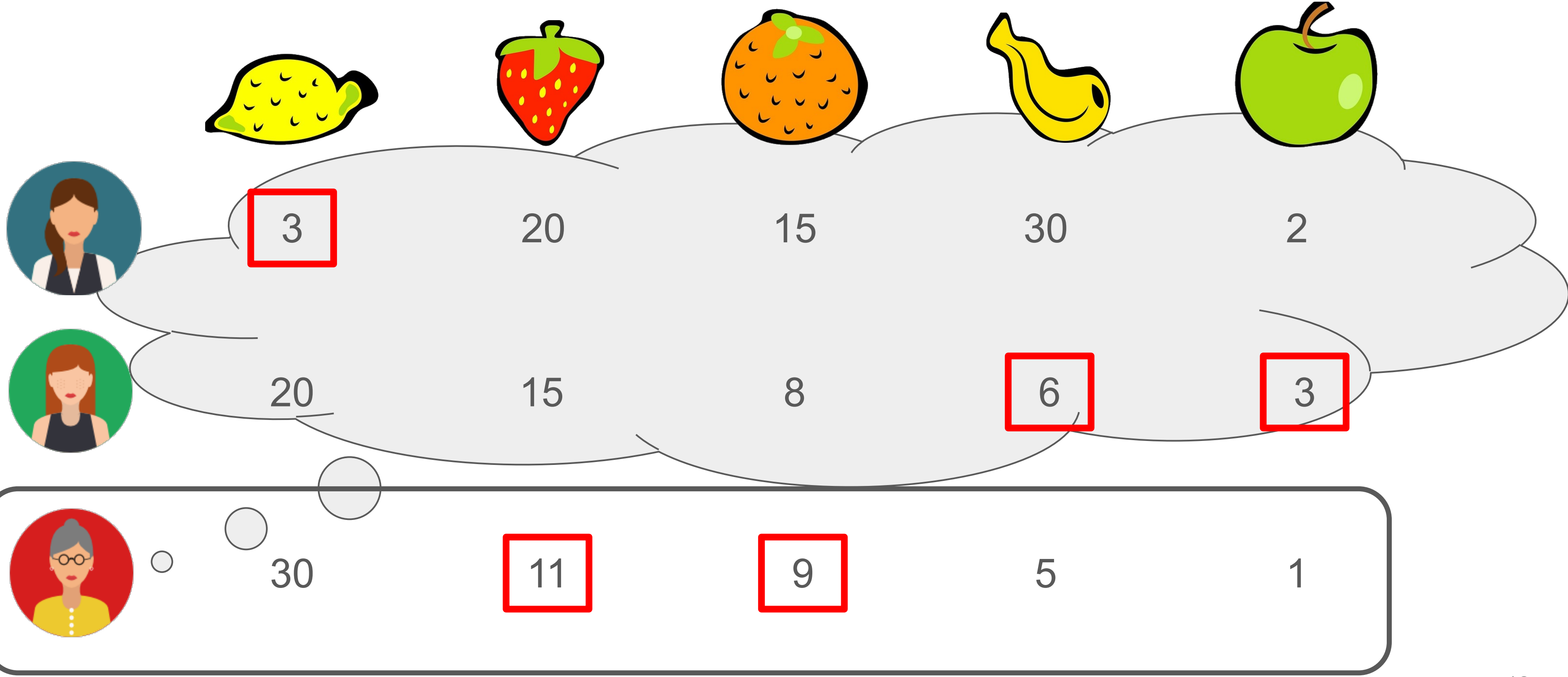
X is EEFX

$\Leftrightarrow$

$\forall\, i \in [n] : \exists\, Y = (Y_1, \ldots, Y_n) :$

- $Y_i = X_i$
- i is EFX satisfied with Y

Red's bundle is such an EEFX certificate



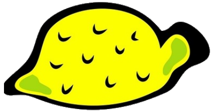
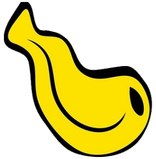
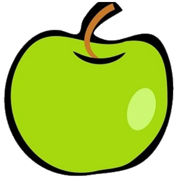
There is a polynomial time algorithm to compute EEFX

Input: instance I = ([n],[m], $\{v_i\}_{i \in [n]}$)

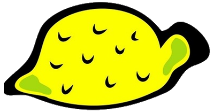
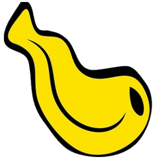
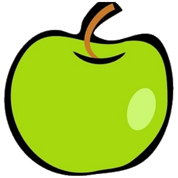
Output: allocation X (EEFX)

1. I’ = ORDER(I)
2. X’ <- ENVY_CYCLE_ELIMINATION(I’)
3. L <- PICKING_SEQUENCE(X’,I’)
4. X <- PICK(I,L)
5. return X

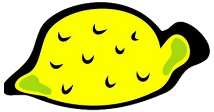
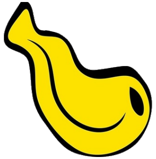
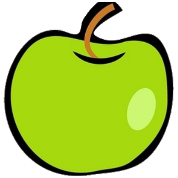
The instance I

					
	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

ORDER(I)

					
	30	20	15	3	2
	30	11	9	5	1
	20	15	8	6	3

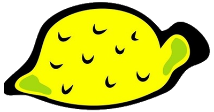
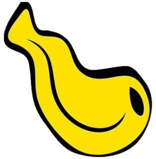
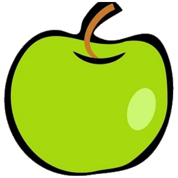
ENVY_CYCLE_ELIMINATION(I’)

					
	30	20	15	3	2
	30	11	9	5	1
	20	15	8	6	3

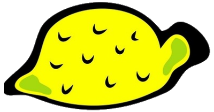
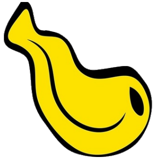
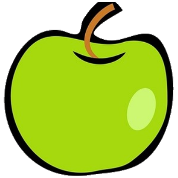
ENVY_CYCLE_ELIMINATION(I’)

	30	20	15	3	2
	30	11	9	5	1
	20	15	8	6	3

ENVY_CYCLE_ELIMINATION(I’)

					
	30	20	15	3	2
	30	11	9	5	1
	20	15	8	6	3

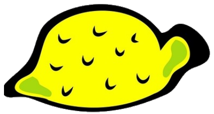
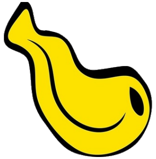
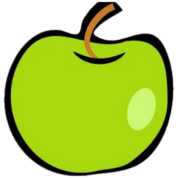
ENVY_CYCLE_ELIMINATION(I')

					
	**30**	20	15	3	2
	30	**11**	9	5	1
	20	15	**8**	6	3

ENVY_CYCLE_ELIMINATION(I’)

	30	20	15	3	2
	30	11	9	5	1
	20	15	8	6	3

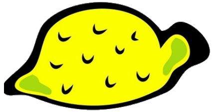
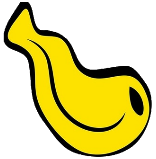
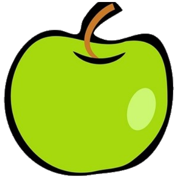
ENVY_CYCLE_ELIMINATION(I’)

					
	**30**	20	15	3	2
	30	11	**9**	**5**	1
	20	**15**	8	6	3

ENVY_CYCLE_ELIMINATION(I')

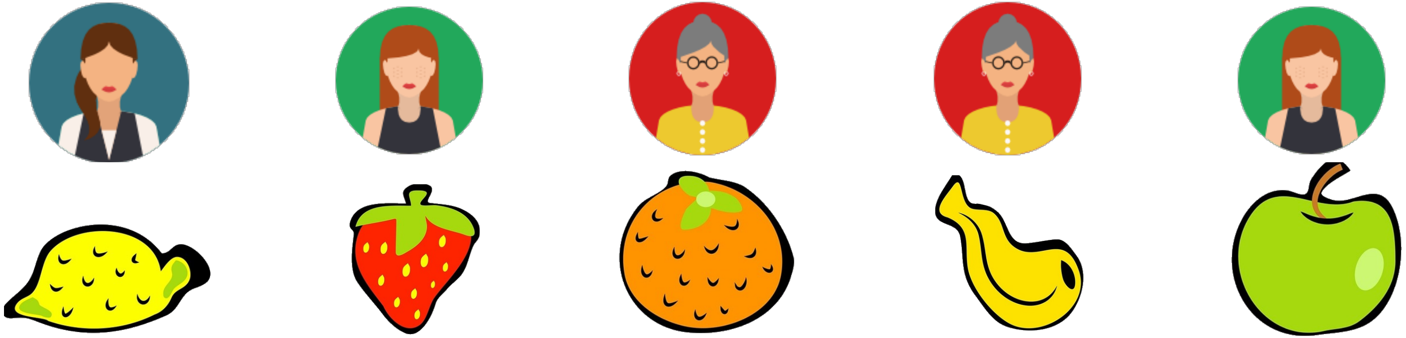
	30	20	15	3	2
	30	11	**9**	**5**	1
	20	**15**	8	6	3

ENVY_CYCLE_ELIMINATION(I’)

					
	**30**	20	15	3	2
	30	11	**9**	**5**	1
	20	**15**	8	6	**3**

This allocation is EFX!

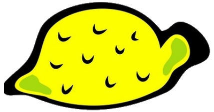
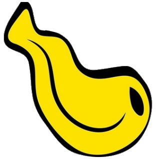
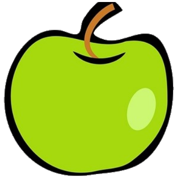
	30	20	15	3	2
	30	11	**9**	**5**	1
	20	**15**	8	6	**3**



30	20	15	3	2
30	11	9	5	1
20	15	8	6	3

PICKING_SEQUENCE(X',I')

PICK(I,L)

					
	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

PICK(I,L)

	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

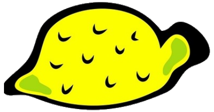
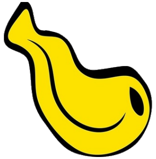
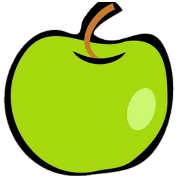
PICK(I,L)

3	20	15	30	2
30	11	9	5	1
20	15	8	6	3

PICK(I,L)

	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

PICK(I,L)

					
	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

PICK(I,L)

	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

The allocation X

3	20	15	**30**	2
30	**11**	**9**	5	1
20	15	8	6	**3**

The algorithm in detail

input: instance $I = ([n],[m], \{v_i\}_{i\in[n]})$

Step 1:
create I'

- enumerate $g_1,g_2,\ldots,g_m$
- define $v_i'(g_j)$ such that g_j is assigned the j highest value out of all the values of v_i

Step 2:
create X'

- run envy cycle elimination on I'

Step 3:
create L

- $L = [L_1,L_2,...,L_m]$
- L_j is the owner of g_j in X'

draft items in I with picking order L

This algorithm is efficient!!

O(m
)

O(m
)

O(m
)

Step 1:
create I'

- enumerate $g_1, g_2, \ldots, g_m$
- define $v_i'(g_j)$ such that g_j is assigned the j highest value out of all the values of v_i

Step 2:
create X'

- run envy cycle elimination on I'

Step 3:
create L

- $L = [L_1, L_2, \ldots, L_m]$
- L_j is the owner of g_j in X'

O(m
)

draft items in I with picking order L

Proof sketch

- Lemma 1 (Plaut and Roughgarden [2020])
  - X' is EFX
- Lemma 2
  - $\forall i \in [n]$: $\exists\ \pi_i$: $[m] \rightarrow [m]$: π_i is a bijection
    - $\forall g \in X_i'$: $\pi_i(g) \in X_i$ and $v_i(\pi_i(g)) \geq v_i'(g)$ (Value does not decrease)
    - $\forall g \notin X_i'$: $\pi_i(g) \notin X_i$ and $v_i(\pi_i(g)) \leq v_i'(g)$ (Value does not increase)
- Lemma 3
  - X (the output) is EEFX

π_i translates goods in X' into goods in X

$g_j \in X_i'$

$\pi_i(g_j)$ is the item picked at time step j of the PICK step

$g_j \notin X_i'$

for the kth item ignoring items picked by agent i: $\pi_i(g_j)$ is the k most valuable item according to i ignoring items picked by agent i

$g_j \in X_i'$

	g_1	g_2	g_3	g_4	g_5
	3	20	15	30	2
		$\pi_i(g_3)$	g_3		
	30	11	9	5	1
	20	15	8	6	3

$g_j \notin X_i'$

	k = 1			k = 2	k = 3
	3	20	15	30	2
	30	11	9	5	1
	20	15	8	6	3

The proof of Lemma 2

Goal:

$\forall i \in [n]: \exists \pi_i: [m]\rightarrow[m] : \pi_i$ is a bijection

- $\forall g \in X_i' : \pi_i(g) \in X_i$ and $v_i(\pi_i(g)) \geq v_i'(g)$
  (Value does not decrease)
- $\forall g \notin X_i' : \pi_i(g) \notin X_i$ and $v_i(\pi_i(g)) \leq v_i'(g)$
  (Value does not increase)

$g_j \in X_i'$

item picked in step j of the algorithm

Set of agent i's j highest valued items:
$G_i^j=\{g_{\sigma i(1)},\ldots,g_{\sigma i(j)}\}$

$\sigma_i : [m] \rightarrow [m]$

- $v_i(g_{\sigma i(j)}) = v_i'(g_j)$
- Random enumeration: $g_1,g_2,\ldots,g_m$
- Ordered enumeration: $g_{\sigma i(1)},\ldots,g_{\sigma i(m)}$

$g_j \notin X_i'$

for the kth item ignoring items picked by agent i: k most valuable item according to i ignoring items picked by agent i

The Proof of Lemma 3

$Y^i = (Y^i_1,...,Y^i_n)$

$Y^i_j = \{\pi_i(g) : g \in X_j'\}$

$g^* \in Y^i_j$ s.t. $g^* = \text{argmin}_g \pi_i(g)$

Claim:

Y^i is an EEFX certificate for agent i

1. $Y^i_i = X_i$
2. agent i is EFX satisfied

$v_i(Y^i_i) \geq v_i(Y^i_j \setminus \{\pi_i(g^*)\}) \quad \forall j \in [m]$

(1) $v_i(Y^i_i) \geq v_i'(X_i')$

(1) $v_i'(X_i') \geq v_i'(X_j' \setminus \{g^*\})$

(1) $v_i'(X_j' \setminus \{g^*\}) \geq v_i(Y^i_j \setminus \{\pi_i(g^*)\})$

MMS implies EEFX

- MMS property: $\forall i : v_i(X_i) \geq \max_Y \min_j v_i(Y_j)$
- "Each agent gets at least the value that is equal to the maximum value of the bundle they receive among all allocations in which they receive their least favourite bundle."

Construct a possible EEFX certificate Y

$Y = (Y_1,\dots,Y_n)$

- $Y_i = X_i$
- $r^{Y,i}$ is lexicograhpically minimum

$r^{Y,i} = (r_1^{Y,i},\dots,r_{n-1}^{Y,i})$

- $r_t^{Y,i} \geq r_{t+1}^{Y,i} \quad \forall t \in [n-2]$
- the entries are the values $v_i(Y_j)$ for all $j \in [n] \setminus \{i\}$

Assume Y is not an EEFX certificate

agent i is not EFX-satisfied in Y

$v_i(Y_i) < v_i(Y_{j1}) - v_i(g)$

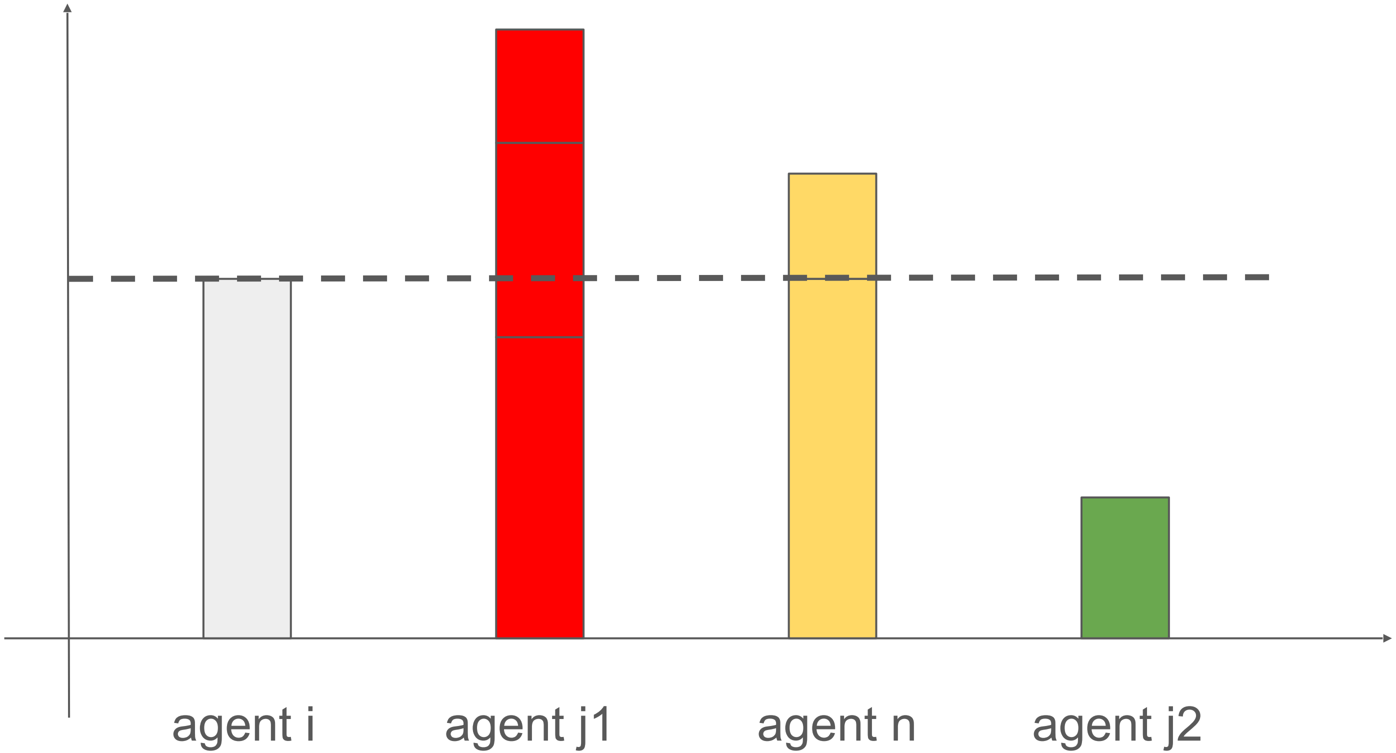
for some other agent j1 and item g

agent i strictly prefers every other bundle

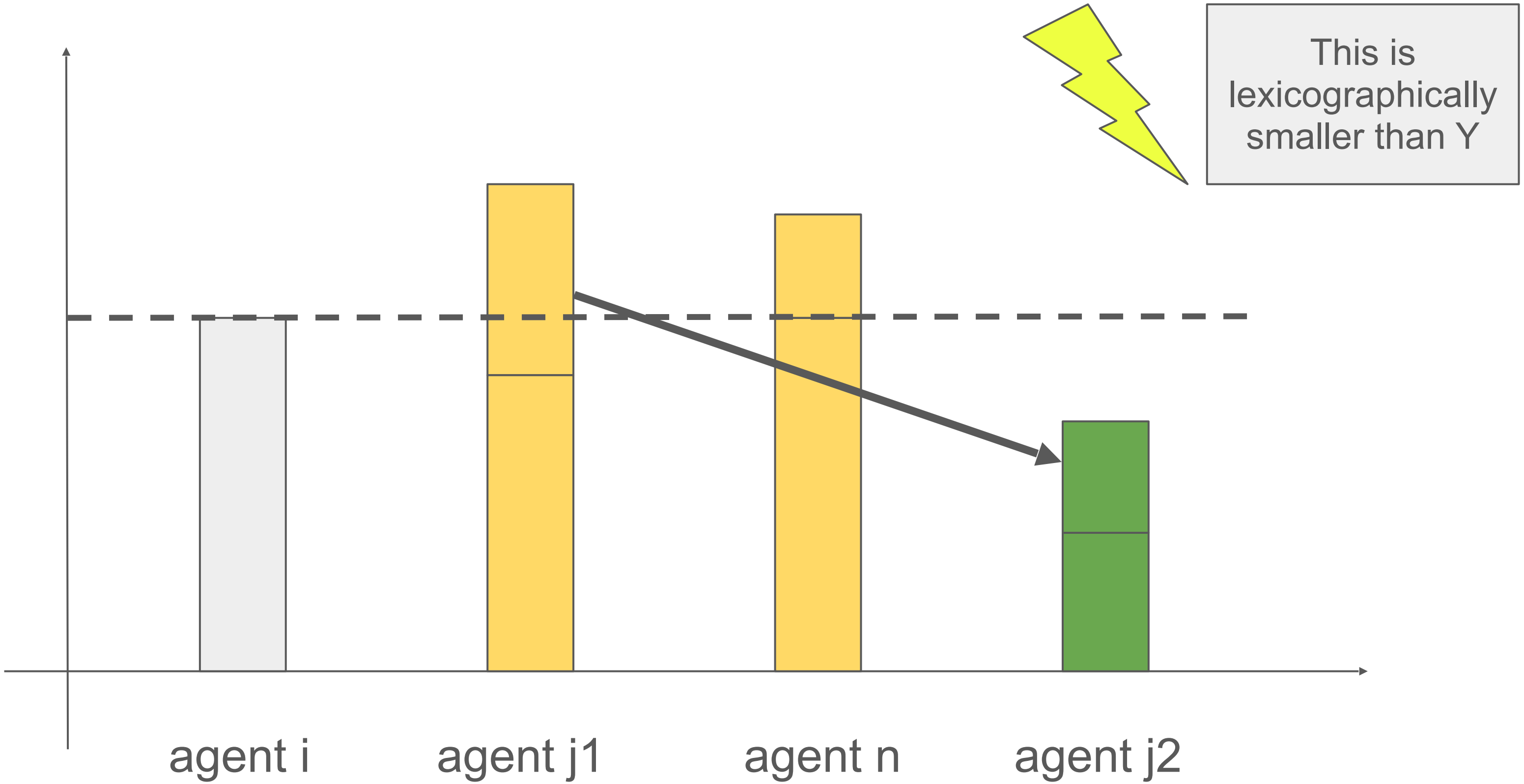
OR

agent i does not prefer at least one of the bundles

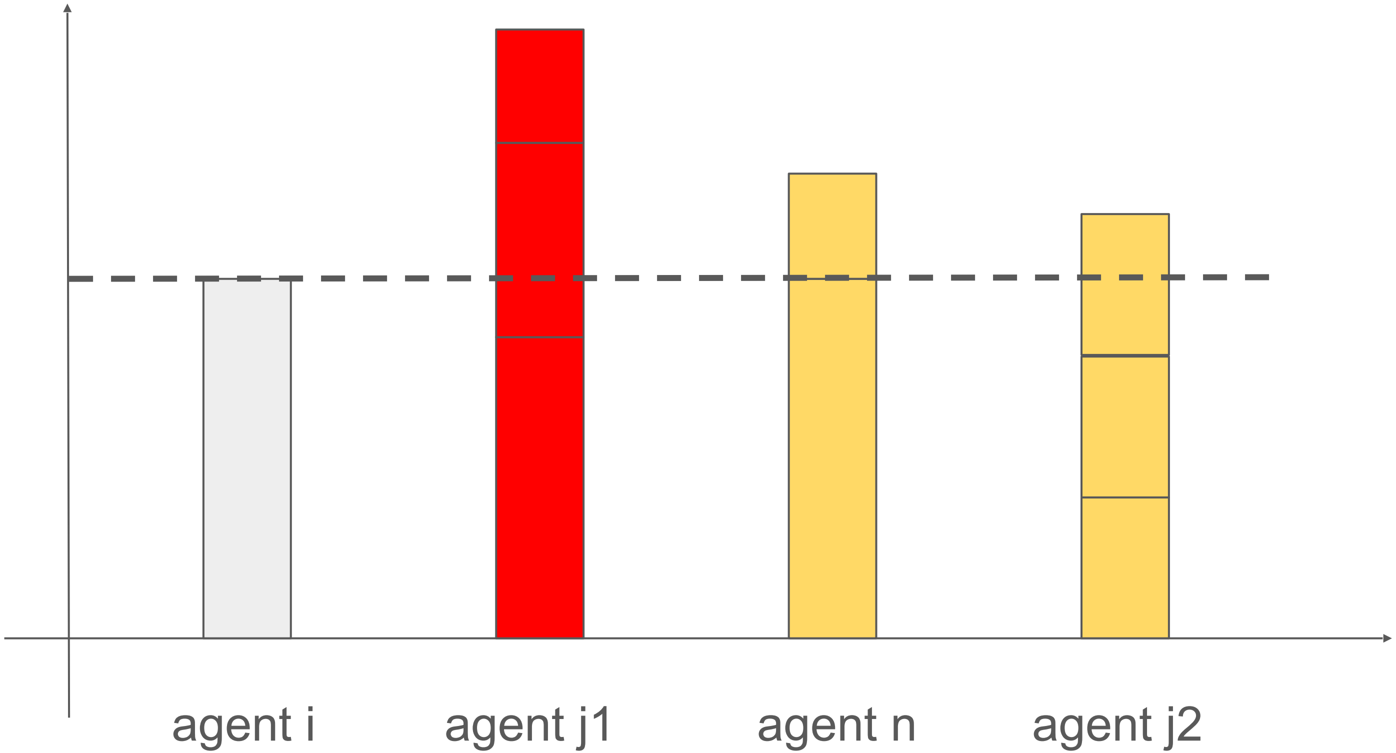
Agent i does not strictly prefer every other bundle



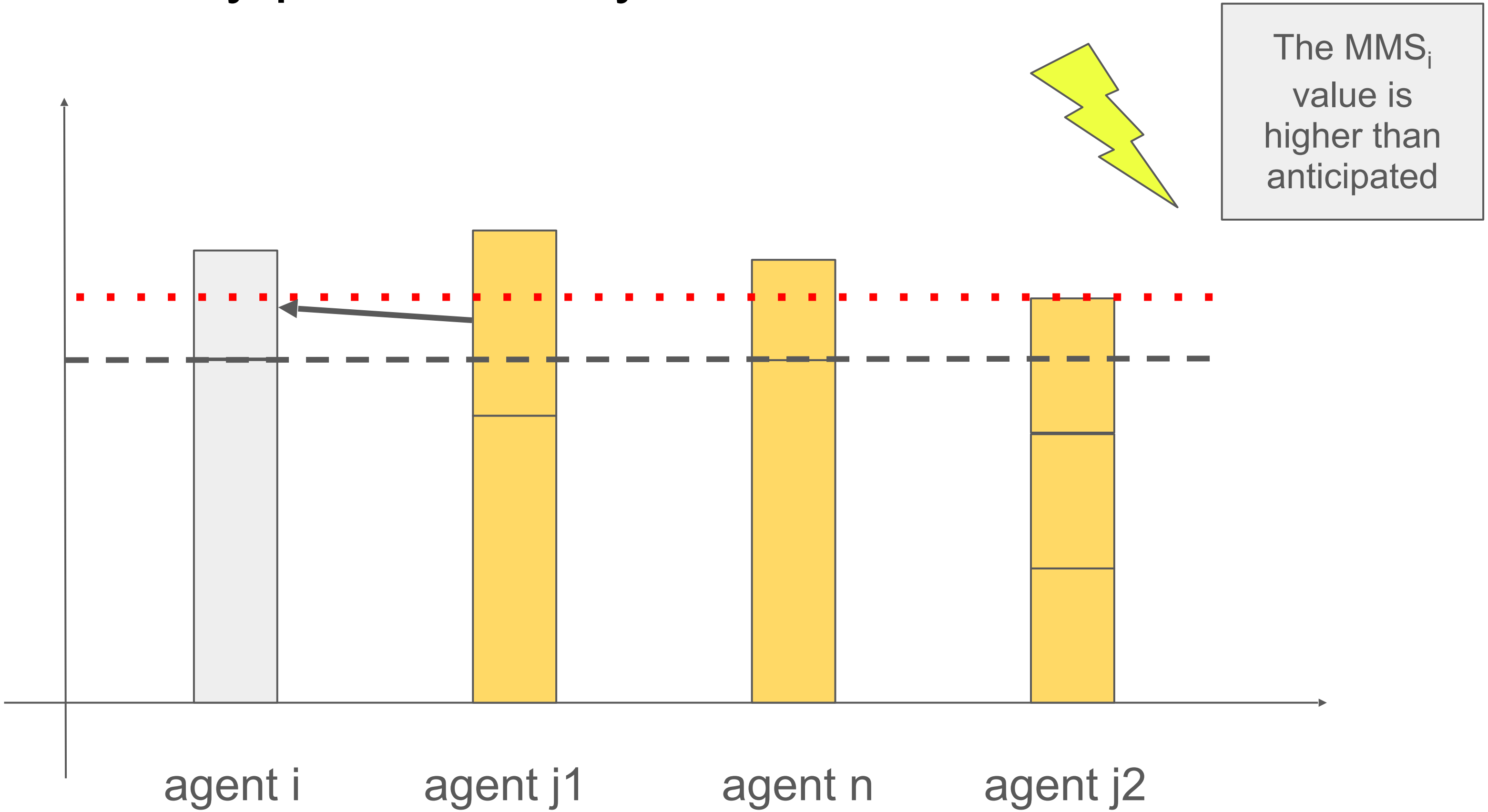
Agent i does not strictly prefer every other bundle



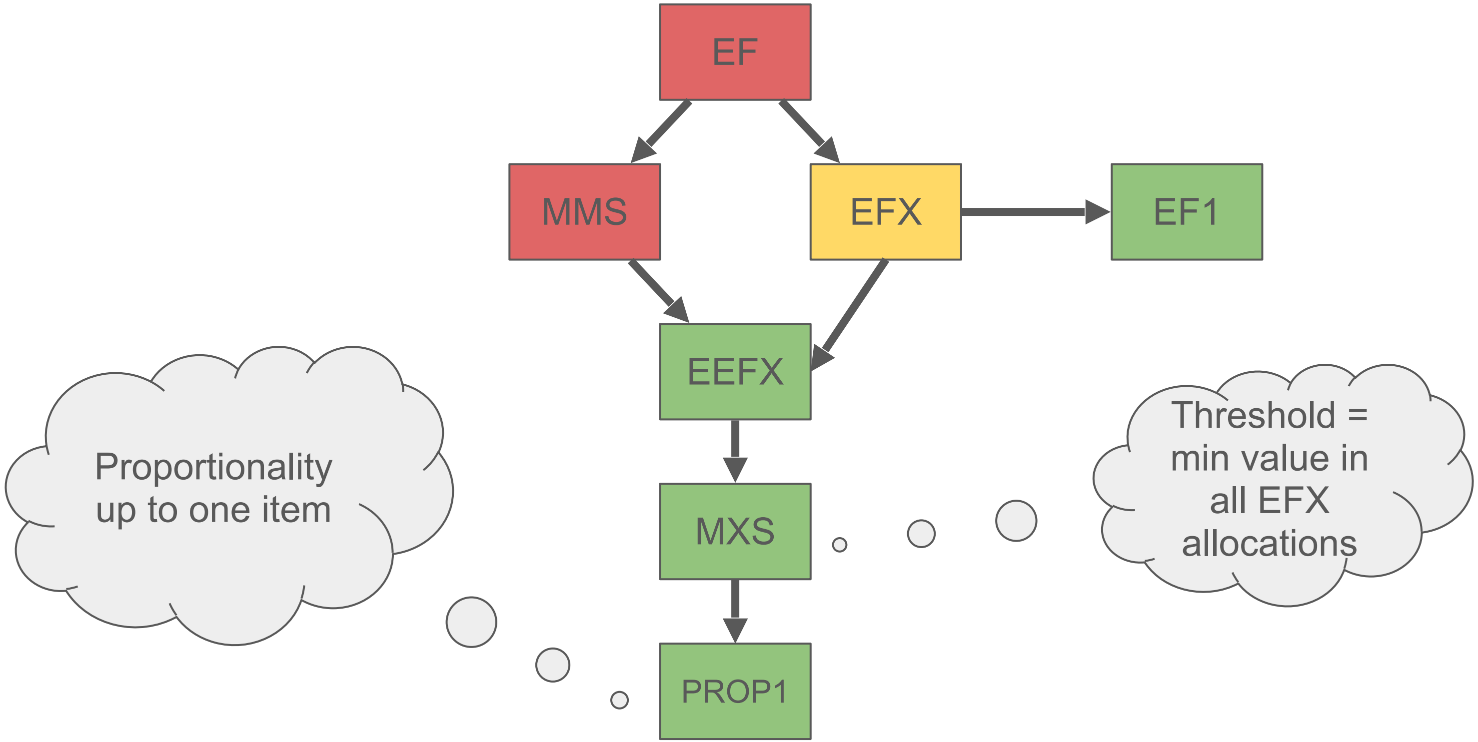
Agent i strictly prefers every other bundle



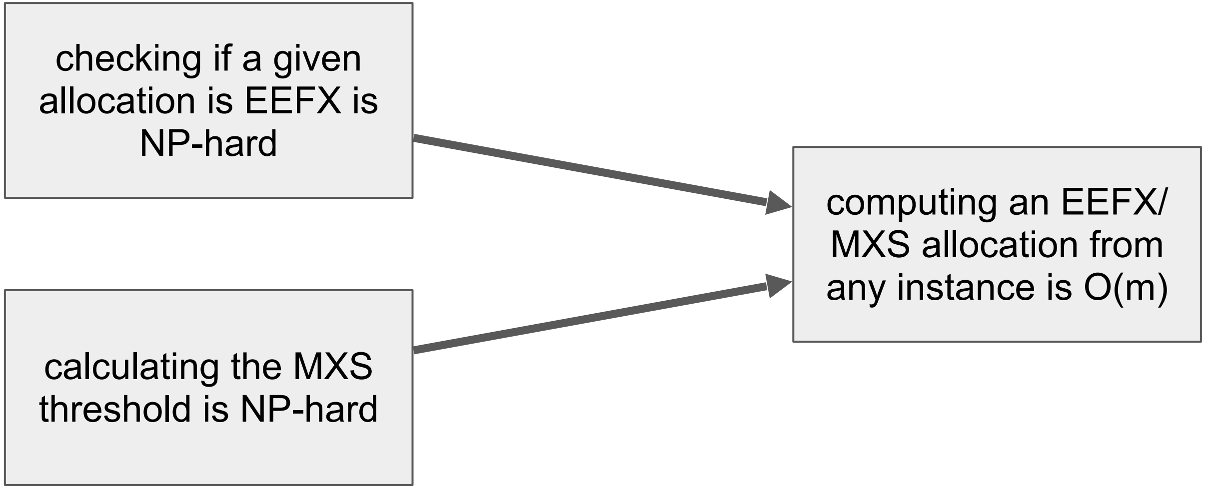
Agent i strictly prefers every other bundle



EEFX fits nicely into the chain of implications



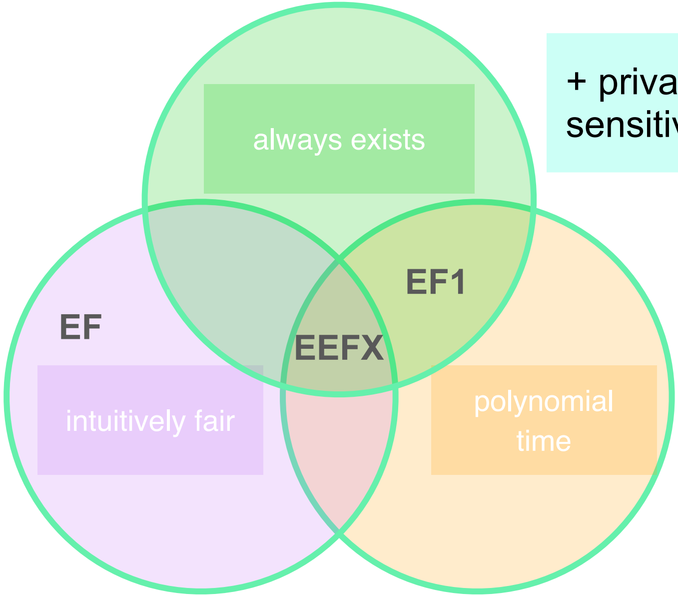
The algorithm is a “short-cut” around a NP-hard problems



Take home messages

EEFX is a new fairness concept that solves previous issues of EF relaxations

MMS and EFX imply EEFX



+ privacy sensitive

MXS is trivially computed by the same algorithm