How to Fairly Allocate Easy and Difficult Chores

Soroush Ebadian[1]

University of Toronto

Dominik Peters

University of Toronto

Nisarg Shah

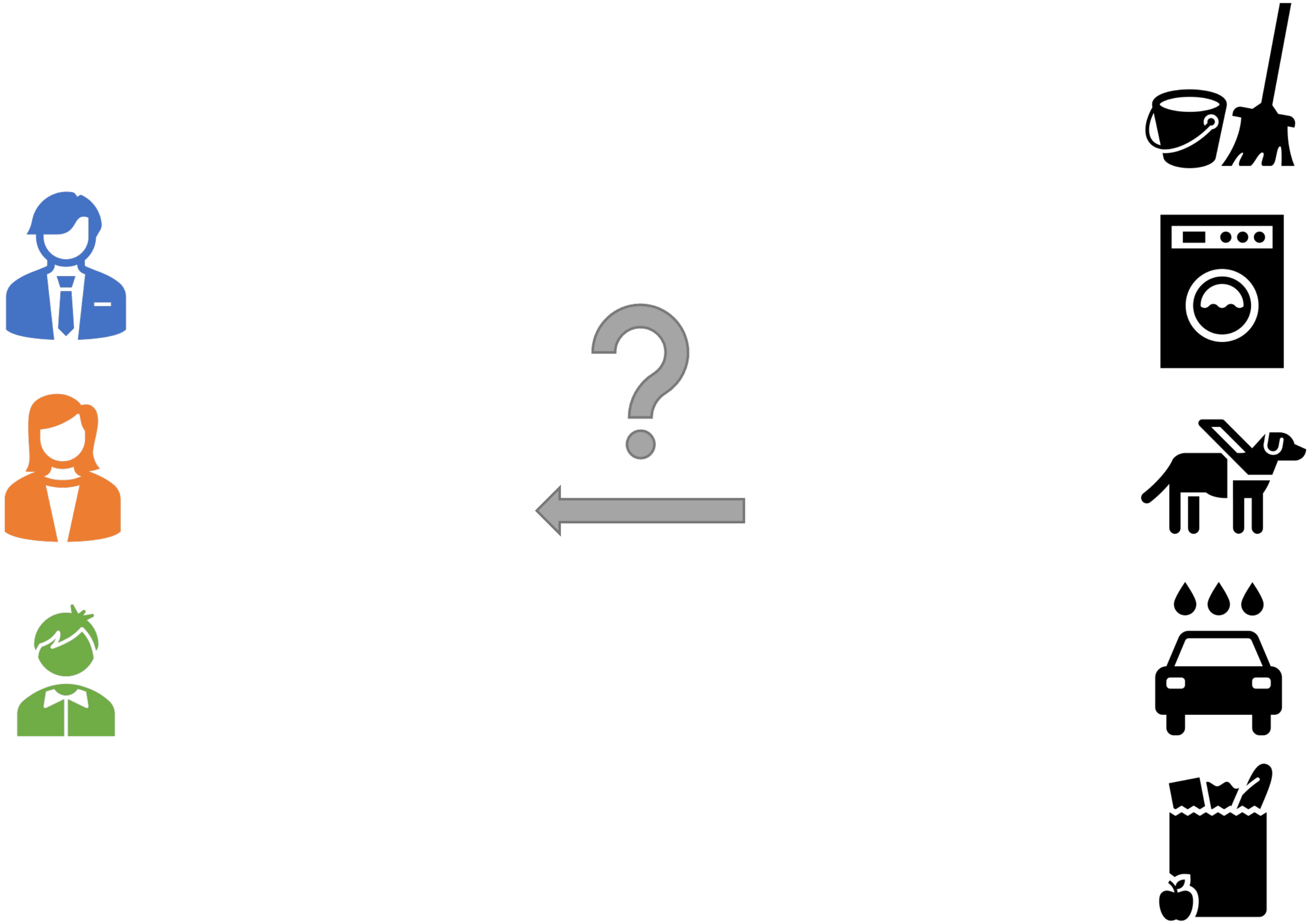
University of Toronto

[1] Email: soroush@cs.toronto.edu

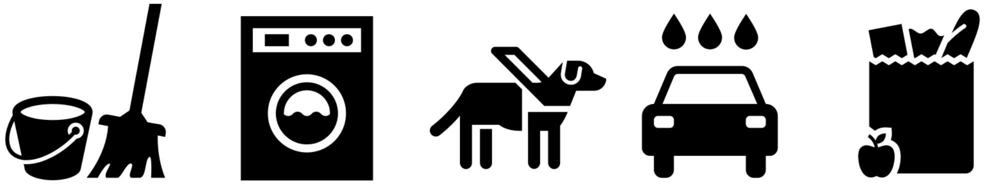
Outline

- Introduction
- Envy-freeness up-to one item + Pareto Optimality
  - Methods for the division of goods
  - Fisher-market-based Algorithms
  - Adapting to chores
- Maximin-Share Fairness

Fair Division of Indivisible Chores

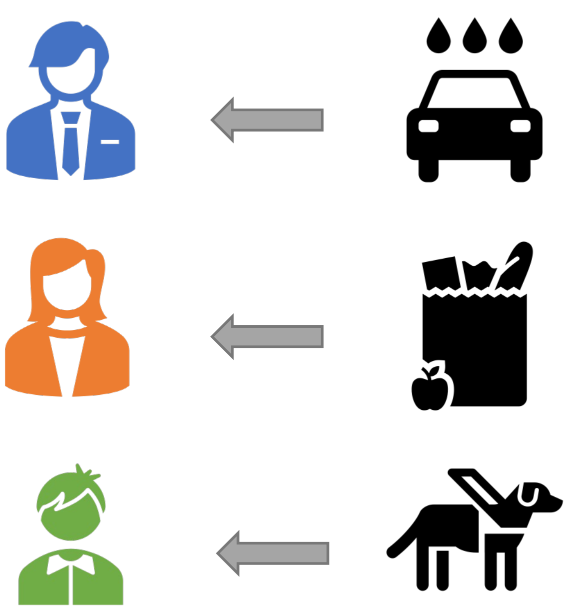


Fair Division of Indivisible Chores



-20	-15	-10	-5	-10
-20	-10	-20	-10	-5
-20	-30	-5	-50	-10

Fair Division of Indivisible Chores



Fair and **Efficient** Allocations

More Formally

- n agents
- m *indivisible* items
- Agent i values item j at $v_{i,j}$
  - **Chores** Instance: $v_{i,j} \in \mathbb{R}_{\leq 0}$
    - Work shifts between staff,
      house chores between roommates, …
- Additive utilities: $v_i(S) = \sum_{j \in S} v_{i,j}$

	-5	-20	-15
	-5	-15	-20

More Formally

- n agents
- m *indivisible* items
- Agent i values item j at $v_{i,j}$
  - **Chores** Instance: $v_{i,j} \in \mathbb{R}_{\leq 0}$
    - Work shifts between staff, house chores between roommates, …
  - **Goods** Instance: $v_{i,j} \in \mathbb{R}_{\geq 0}$
    - Estate (inheritance) division, divorce settlement, …
- Additive utilities: $v_i(S) = \sum_{j \in S} v_{i,j}$

	-5	-20	-15
	-5	-15	-20

	5	20	15
	10	20	10

More Formally

- n agents
- m *indivisible* items
- Agent i values item j at $v_{i,j}$
  - **Chores** Instance: $v_{i,j} \in \mathbb{R}_{\leq 0}$
    - Work shifts between staff, house chores between roommates, …
  - **Goods** Instance: $v_{i,j} \in \mathbb{R}_{\geq 0}$
    - Estate (inheritance) division, divorces settlement, …
- Additive utilities: $v_i(S) = \sum_{j \in S} v_{i,j}$

-5	-20	-15
-5	-15	-20

5	20	15
10	20	10

Goal: Find an **allocation** $A = (A_1, A_2, \ldots, A_n)$ that is ***fair*** and ***efficient***.

Gold Standard Fairness Notion

- Envy-Freeness (EF):
  - No agent prefers another one’s bundle to their allocated bundle.
  - I.e., for all pairs of agents i, j:

$$v_i(A_i) \geq v_i(A_j)$$

Gold Standard Fairness Notion

- Envy-Freeness (EF):
  - No agent prefers another one's bundle to their allocated bundle.
  - I.e., for all pairs of agents i, j:

$$v_i(A_i) \geq v_i(A_j)$$

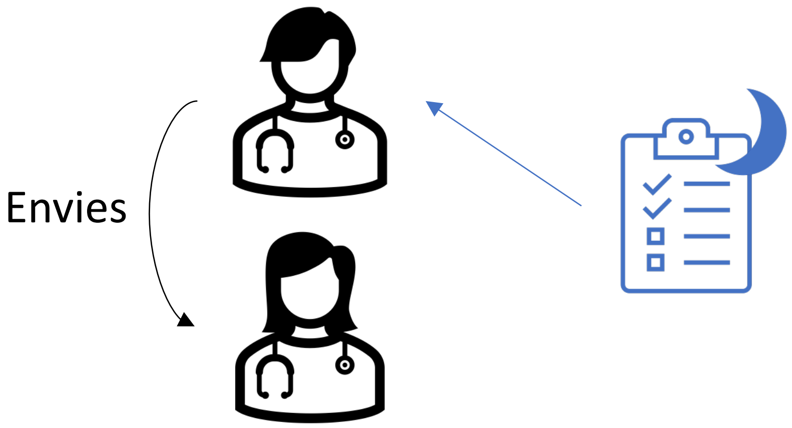
Envy-free allocations may not exist.

Gold Standard Fairness Notion

- Envy-Freeness (EF):
  - No agent prefers another one's bundle to their allocated bundle.
  - I.e., for all pairs of agents i, j:

$$v_i(A_i) \geq v_i(A_j)$$

Envy-free allocations may not exist.

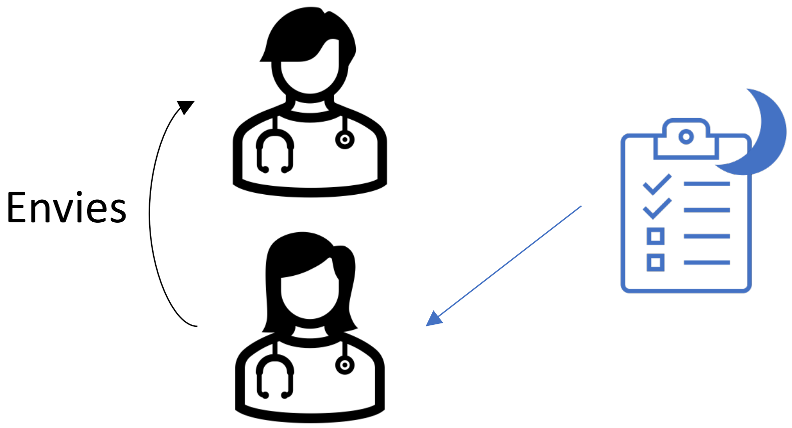


Gold Standard Fairness Notion

- Envy-Freeness (EF):
  - No agent prefers another one's bundle to their allocated bundle.
  - I.e., for all pairs of agents i, j:

$$v_i(A_i) \geq v_i(A_j)$$

Envy-free allocations may not exist.



Relaxed Fairness Notion

- Envy-Freeness up to one item (EF1):
  - No agent prefers another one's bundle to their allocated bundle, **after ignoring at most one item.**
  - **Chores** Instance:
    For all pairs of agents i, j:

$$\exists c \in A_i : \quad v_i(A_i \setminus \{c\}) \geq v_i(A_j)$$

Relaxed Fairness Notion

- Envy-Freeness up to one item (EF1):
  - No agent prefers another one's bundle to their allocated bundle, **after ignoring at most one item.**
  - **Chores** Instance:
    For all pairs of agents i, j:
    $$\exists c \in A_i : \quad v_i(A_i \setminus \{c\}) \geq v_i(A_j)$$
  - **Goods** Instance:
    For all pairs of agents i, j:
    $$\exists g \in A_j : \quad v_i(A_i) \geq v_i(A_j \setminus \{g\})$$

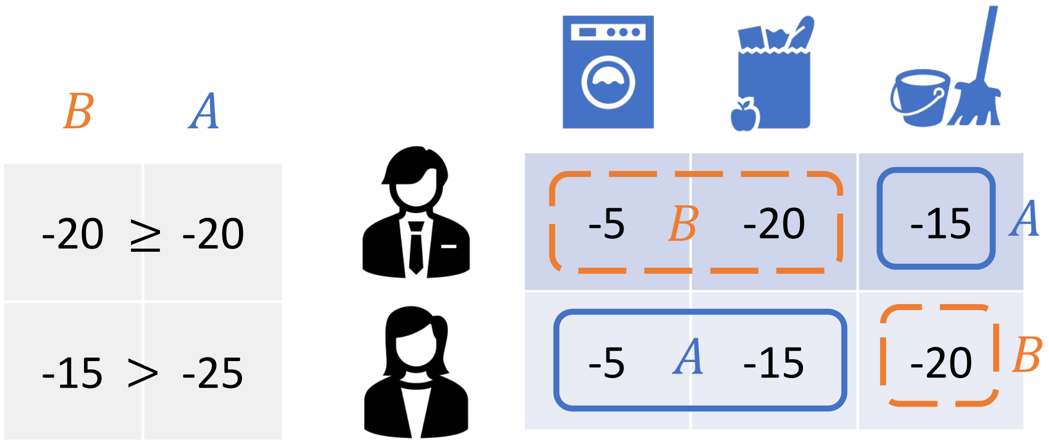
Relaxed Fairness Notion

- Envy-Freeness up to one item (EF1):
  - No agent prefers another one's bundle to their allocated bundle, **after ignoring at most one item.**
  - **Chores** Instance:
    For all pairs of agents i, j:
    $$\exists c \in A_i : \quad v_i(A_i \setminus \{c\}) \geq v_i(A_j)$$
  - **Goods** Instance:
    For all pairs of agents i, j:
    $$\exists g \in A_j : \quad v_i(A_i) \geq v_i(A_j \setminus \{g\})$$

EF1 allocations ***always*** exist.

Efficiency Notion

- Pareto Optimality (PO):
  - Allocation A is Pareto optimal,
    if there is no allocation B such that $\forall i:\ \ v_i(B) \geq v_i(A)$,
    and $\exists j:\ \ v_j(B) > v_j(A)$.



A is **not** Pareto Optimal as B Pareto dominates it.

Fair and Efficient Allocations

Does a ***fair*** (EF1) and ***efficient*** (PO) allocation always exist?

Fair and Efficient Allocations

Does a *fair* **(EF1)** and *efficient* **(PO)** allocation always exist?

Goods

- EF1 + PO allocations always exist. (Caragiannis et al., 2016)
- Can be found in pseudo-polynomial time. (Barman et al., 2018)
- Poly-time when utility levels are poly-sized / constantly many agents. (Garg et al., 2021)

Fair and Efficient Allocations

Does a *fair* **(EF1)** and *efficient* **(PO)** allocation always exist?

Open Problem 1.
Can EF1+PO allocations be found in poly time?

Goods

- EF1 + PO allocations always exist.
  (Caragiannis et al., 2016)
- Can be found in pseudo-polynomial time. (Barman et al., 2018)
- Poly-time when utility levels are poly-sized / constantly many agents.
  (Garg et al., 2021)

Fair and Efficient Allocations

Does a *fair* **(EF1)** and *efficient* **(PO)** allocation always exist?

Chores

- Still open for additive valuations.

Our key contribution:

Theorem 1. For **Bivalued** chores, EF1 + PO allocations **always exist**, and can be found in **poly time.**

Bivalued utilities:

$$\forall i, j\colon\ v_{i,j} \in \{a, b\},$$
$$a \leq b \leq 0,$$

Goods

- EF1 + PO allocations always exist. (Caragiannis et al., 2016)
- Can be found in pseudo-polynomial time. (Barman et al., 2018)
- Poly-time when utility levels are poly-sized / constantly many agents. (Garg et al., 2021)

Results and Techniques for Goods

- Solution 1: Maximizing Nash Welfare (MNW)
  i.e., $\max_A \prod_i v_i(A_i)$ or $\max_A \sum_i \log(v_i(A_i))$.
  - Goods: MNW yields EF1 + PO (Caragiannis et al., 2016)

Results and Techniques for Goods

- Solution 1: Maximizing Nash Welfare (MNW)

  i.e., $\max_A \prod_i v_i(A_i)$ or $\max_A \sum_i \log(v_i(A_i))$.

  - Goods: MNW yields EF1 + PO (Caragiannis et al., 2016)

- For integral utilities:

  Maximizing Harmonic Welfare yields EF1 + PO

  i.e., $\max_A \sum_i H\big(v_i(A_i)\big)$ (Montanari et al., 2022)

  - $H(i) = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{i}$

Results and Techniques for Goods

- Solution 1: Maximizing Nash Welfare (MNW)

  i.e., $\max_{A} \prod_i v_i(A_i)$.

  - Goods: MNW yields EF1 + PO (Caragiannis et al., 2016)

- Chores: (1) Maximizing $\prod_i |v_i(A_i)|$? or $\prod_i (v_i(A) - v_i(A_i))$

  No, favors higher disutilities. No, counter example.

  (2) Maximizing $\prod_i |v_i(A_i)|$ subject to PO?

  No, fails EF1. (Example with bivalued utilities, n=4, m=8)

  (3) Minimizing $\prod_i |v_i(A_i)|$?

  No, favors having an idle agent with no tasks.

Results and Techniques for Goods

- Solution 1: Maximizing Nash Welfare (MNW)
  - Extension to Chores: not obvious.

Results and Techniques for Goods

- Solution 1: Maximizing Nash Welfare (MNW)
  - Extension to Chores: not obvious.

- Solution 2: Fisher market adaptation (Barman et al. (2018))
  - Finds an EF1 + PO allocation in pseudo-poly time
  - Idea: a *local search* terminates due to *invariants* and potential functions

Results and Techniques for Goods

- Solution 1: Maximizing Nash Welfare (MNW)
  - Extension to Chores: not obvious.

- Solution 2: Fisher market adaptation (Barman et al. (2018))
  - Finds an EF1 + PO allocation in pseudo-poly time
  - Idea: a *local search* terminates due to *invariants* and potential functions

  - Extension to Chores:
    - Non-trivial, *invariants* cease to hold
    - Our result:
      With a more intricate analysis → EF1 + PO for **Bivalued** Utilities

Fisher Markets in Fair Division

Fisher Markets

Setup:

- n agents, m items
- Item prices: $p_j \in \mathbb{R}_{\geq 0}$

	\$2	\$10	\$10
	-3 **BB: -1.5**	-20 BB: -2	-15 **BB: -1.5**
	-3 **BB: -1.5**	-15 **BB: -1.5**	-20 BB: -2

Def. Bang per Buck: $\frac{v_{i,j}}{p_j}$

Maximum Bang per Buck: $\mathrm{MBB}_i = \max_j \frac{v_{i,j}}{p_j}$

Fisher Markets

Setup:

- n agents, m items
- Item prices: $p_j \in \mathbb{R}_{\geq 0}$

		$2	$10	$10
MBB = -1.5		-3 **<u>BB: -1.5</u>**	-20 BB: -2	-15 **<u>BB: -1.5</u>**
MBB = -1.5		-3 **<u>BB: -1.5</u>**	-15 **<u>BB: -1.5</u>**	-20 BB: -2

Def. Bang per Buck: $\frac{v_{i,j}}{p_j}$

Maximum Bang per Buck: $\mathrm{MBB}_i = \max_j \frac{v_{i,j}}{p_j}$

Equilibrium:

- All items are allocated
- Agents are only allocated MBB items

First Welfare Theorem → **Pareto Optimal (PO)**

Fisher Markets

- An allocation is price envy-free up to one item (**pEF1**) if for all pairs of agents i, j:

$$\exists c \in A_j : \;\; p\left(A_j \setminus \{c\}\right) \leq p(A_i)$$

pEF1 + equilibrium $\Longrightarrow$ **EF1 + PO**

Fisher Markets

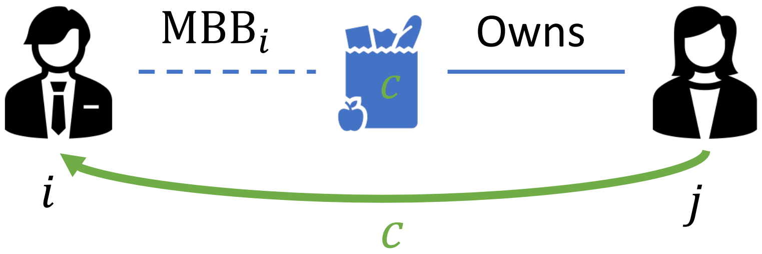
- An allocation is price envy-free up to one item (**pEF1**) if for all pairs of agents i, j:

$$\exists c \in A_j : \; p\big(A_j \setminus \{c\}\big) \leq p(A_i)$$

pEF1 + equilibrium → **EF1 + PO**

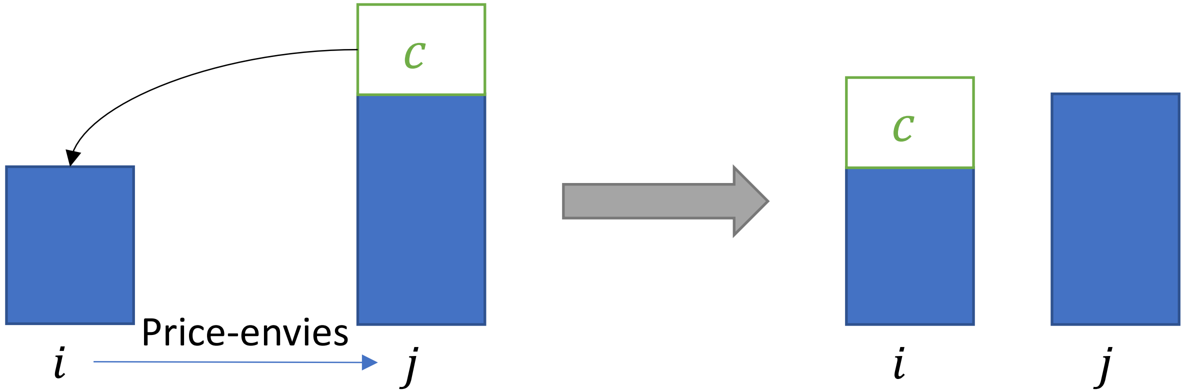
- Algorithmic Framework:
  - Start with an allocation and prices in equilibrium
  - Make local changes reducing envy (while remaining in equilibrium)
  - Reach pEF1 (+ equilibrium)

Fisher Market Algorithm Ideas



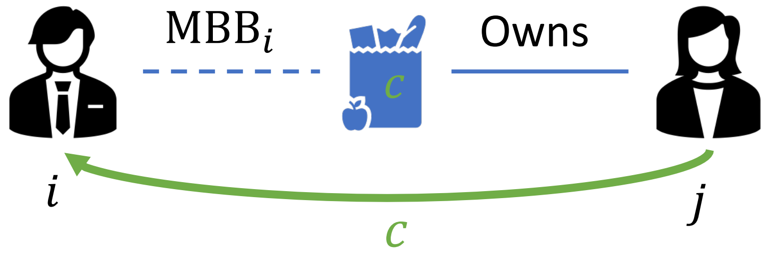
- MBB Graph
  - Edge $i \overset{c}{\leftarrow} j$
- Local changes:

Suppose $i \overset{c}{\leftarrow} j$ exists and $p(A_i) < p(A_j) - p_c$ (violation of pEF1),
then transferring c to i_1 (1) remains in the equilibrium
(2) reduces envy "overall"



Fisher Market Algorithm for Goods

- MBB Graph
  - Edge $i \overset{c}{\leftarrow} j$



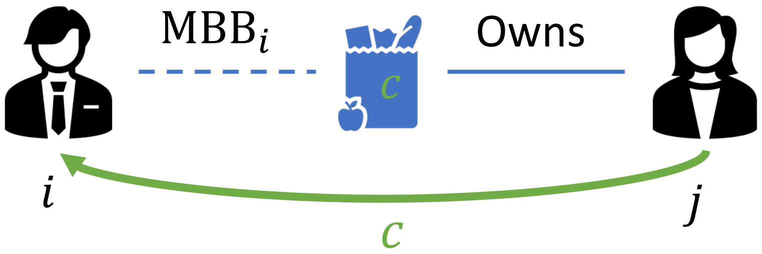
- Algorithm Sketch for Goods (Barman et al. 2018)
  1. Start with welfare maximizing allocation
  2. Least Spender: $ls = \operatorname{argmin}_i p(A_i)$
  3. While there is $ls \overset{c_1}{\leftarrow} i_2 \overset{c_2}{\leftarrow} i_3 \overset{c_3}{\leftarrow} \ldots \overset{c_{\ell-1}}{\longleftarrow} i_\ell$ where ls price envies i_ℓ:

     Take the shortest path, **make a local transfer**, go to 2.
  4. If not pEF1:

     Raise prices of items allocated to ls and **agents reaching ls**, go to 2.

Fisher Market Algorithm for Goods

- MBB Graph
  - Edge $i \overset{c}{\leftarrow} j$



- Algorithm Sketch for Goods (Barman et al. 2018)
  1. Start with welfare maximizing allocation
  2. Least Spender: $ls = \operatorname*{argmin}_i p(A_i)$
  3. While there is $ls \overset{c_1}{\leftarrow} i_2 \overset{c_2}{\leftarrow} i_3 \overset{c_3}{\leftarrow} \ldots \overset{c_{\ell-1}}{\longleftarrow} i_\ell$ where ls price envies i_ℓ:

     Take the shortest path, **make a local transfer**, go to 2.
  4. If not pEF1:

     Raise prices of items allocated to ls and **agents reaching ls**, go to 2.

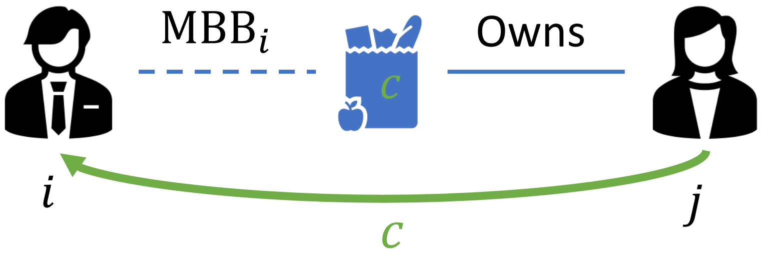
Key Invariants:

$\downarrow \max_i \min_{c \in A_i} p(A_i) - p(c)$

$\uparrow \min_i p(A_i)$

Attempt 1: Algorithm for *Chores*

- MBB Graph
  - Edge $i \xleftarrow{c} j$



- Sketch of Adaptation for Chores
  1. Start with welfare maximizing allocation
  2. Least Spender: $ls = \mathrm{argmin}_i\, p(A_i)$
  3. While there is $ls \xleftarrow{c_1} i_2 \xleftarrow{c_2} i_3 \xleftarrow{c_3} \ldots \xleftarrow{c_{\ell-1}} i_\ell$ where ls price envies i_ℓ:
     Take the shortest path, **make a local transfer**, go to 2.
  4. If not pEF1:
     ~~Raise~~ Reduce prices of items allocated to ls and **agents reaching** ls, go to 2.

Known invariants and potential functions break.

Key Invariants:

$\downarrow \max_i \min_{c \in A_i} p(A_i) - p(c)$

$\uparrow \min_i p(A_i)$

Algorithm for **Bivalued** Chores

[Phase 1: Init]

1. Start with welfare maximizing allocation

[Phase 2a]

2.

[Phase 2b: Reallocate chores]

3. Least Spender: $ls = \text{argmin}_i\, p(A_i)$
4. While there is $ls \xleftarrow{c_1} i_2 \xleftarrow{c_2} i_3 \xleftarrow{c_3} \dots \xleftarrow{c_{\ell-1}} i_\ell$
   where ls price envies i_ℓ after removing $c_{\ell-1}$:
   Take the shortest path, make a local transfer, go to 3.

[Phase 3: Price Reduction]

5. If not pEF1:
   Reduce prices of items allocated to ls and agents reaching ls (H_k), go to 2.

Algorithm for **Bivalued** Chores

[Phase 1: Init]

1. Start with welfare maximizing allocation, k = 0

[Phase 2a]

2. Eliminate price envy between H_k's, k = k + 1

[Phase 2b: Reallocate chores]

3. Least Spender: $ls = \text{argmin}_i \, p(A_i)$
4. While there is $ls \xleftarrow{c_1} i_2 \xleftarrow{c_2} i_3 \xleftarrow{c_3} \dots \xleftarrow{c_{\ell-1}} i_\ell$
   where ls price envies i_ℓ after removing $c_{\ell-1}$:
   Take the shortest path, make a local transfer, go to 3.

[Phase 3: Price Reduction]

5. If not pEF1:
   Reduce prices of items allocated to ls and agents reaching ls, go to 2.

Algorithm for **Bivalued** Chores

[Phase 1: Init]

1. Start with welfare maximizing allocation, k = 0

[Phase 2a]

2. Eliminate price envy between H_k's, k = k + 1

[Phase 2b: Reallocate chores]

3. Least Spender: $ls = \text{argmin}_i\, p(A_i)$
4. While there is $ls \xleftarrow{c_1} i_2 \xleftarrow{c_2} i_3 \xleftarrow{c_3} \ldots \xleftarrow{c_{\ell-1}} i_\ell$
   where ls price envies i_ℓ after removing $c_{\ell-1}$:
   Take the shortest path, make a local transfer, go to 3.

[Phase 3: Price Reduction]

5. If not pEF1:
   Reduce prices of items allocated to ls and agents reaching ls, go to 2.

Key Idea:

We can make H_k's **disjoint**.

Each agent experiences price reduction at most once.
$\Rightarrow$ At most n Phase 3's

Proof by **Induction.**

So far

- EF1 + PO allocations always exist for **bivalued** chores
- Major open problems:
  - Complexity of EF1 + PO allocations for goods?
  - Does EF1 + PO allocations always exist for chores?
- Chores division seems *harder* than Goods division

Maximin Share Fairness

Another Fairness Notion

- Maximin Share (MMS) Allocation (Budish, 2011)
  - For all agents i,

$$v_i(A_i) \geq \mathrm{MMS}_i \quad \text{(MMS value)}$$

  - MMS value of agent i:

$$\mathrm{MMS}_i = \max_{P=(P_1,P_2,\ldots,P_n)} \min_{P_j \in P} v_i(P_j)$$

Another Fairness Notion

- Maximin Share (MMS) Allocation (Budish, 2011)
  - For all agents i,

$$v_i(A_i) \geq \mathrm{MMS}_i \quad \text{(MMS value)}$$

  - MMS value of agent i:

$$\mathrm{MMS}_i = \max_{P=(P_1,P_2,\ldots,P_n)} \min_{P_j \in P} v_i(P_j)$$

Finding MMS values is **NP-hard**.

MMS Allocations

- MMS allocations may **not** exist in general. (Procaccia et al. (2014), Kurokowa et al. (2016))
  - Approach 1: Approximation results for general instances (Huang and Lu (2021), Garg and Taki (2020), …)
  - Approach 2: Existential results for ***restricted*** instances
    - Binary: $v_{i,j} \in \{0, 1\}$ or $v_{i,j} \in \{0, -1\}$
    - Ternary: $v_{i,j} \in \{0, 1, 2\}$ (Amanatidis et al. (2017))
    - Lexicographical (Hosseini et al. (2018))
    - Two other classes of utilities (This work)

Factored Valuations

- Factored valuations:

$$v_{i,j} \in \{0, p_1, p_2, \dots, p_k\} \mid p_\ell = p_{\ell-1} \cdot q \text{ , for some } q \in \mathbb{N}.$$

Lemma. For *factored* valuations, **MMS value** and a corresponding partition can be found in **poly-time**.

Personalized *Factored* Bivalued

- Personalized Bivalued: $v_{i,j} \in \{a_i, b_i\}$
- Factored: $\frac{a_i}{b_i} \in \mathbb{N}$

Theorem 2 (a). For personalized *factored* bivalued **chores** or **goods**:

- MMS allocation always exist
- **MMS + PO** allocation can be found in **poly time**

- Feige (2022): MMS exists for bivalued utilities (non-personalized).

Weakly Lexicographic Preferences

- Agents rank items by undesirability allowing ties

  Undesirability levels: $\{a \sim b \sim d\} \succ \{e \sim f\} \succ \{g \sim h \sim k\}$

- Ties within a level: $c \sim c' \rightarrow |v_{i,c}| = |v_{i,c'}|$

  E.g. $v_{i,a} = v_{i,b}$

- Lexicographic preference between levels: $|v_{i,c}| > \sum_{c' \prec c} |v_{i,c'}|$

  E.g. $|v_i(a)| > |v_i(\{e, f, g, h, k\})|$

Theorem 2 (b). For weakly lexicographic **chores** or **goods:**

- MMS allocation always exist
- **MMS + PO** allocation can be found in **poly time**

Conclusion and Future Work

- EF1 + PO exists for bivalued chores
  - Chores seem harder than Goods
- MMS exists for two subclasses of factored utilities
  - Weakly lexicographic, Personalized factored bivalued

- **Open questions**
  - EFX + PO for bivalued?

    EFX: no envy after removing **any** chore
  - EF1 + PO for trivalued or weakly lexicographic instances?
  - MMS for factored valuations?

Acknowledgements

Our EF1 + PO result was recently independently obtained by a AAAI paper, using a similar technique (Garg et al. (2022))

Thank you!