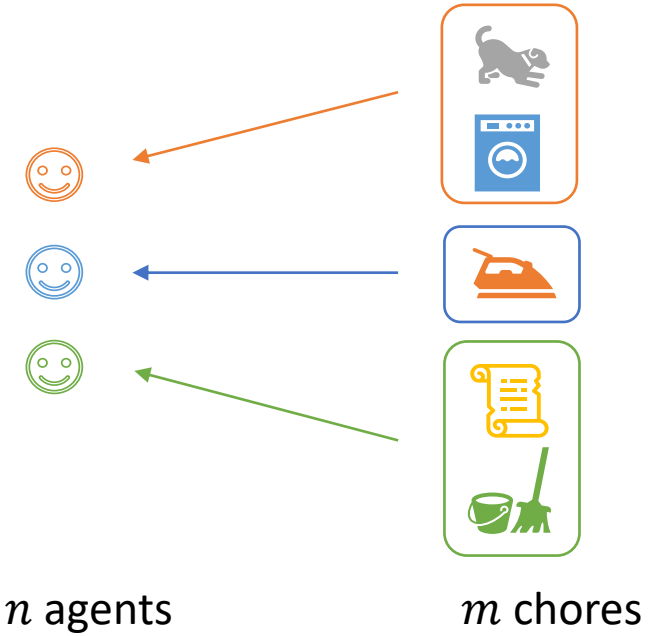
New Algorithms for the Fair and Efficient Allocation of Indivisible Chores

Jugal Garg, **Aniket Murhekar**, John Qin

University of Illinois, Urbana-Champaign (UIUC)

IJCAI/2023 MACAO

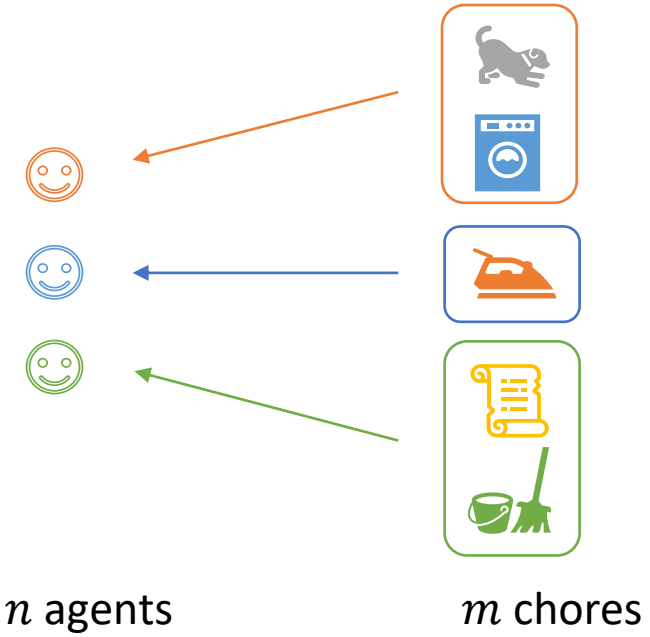
Fair Division of Indivisible Chores



- Goods: Items that provide value/utility to agents receiving them
- Chores: Items that impose a cost/disutility on agents receiving them
- Problem: Allocate chores to agents in a fair and efficient manner
- Many practical applications:
  - Partnership dissolution
  - Allocating tasks to machines

The Model

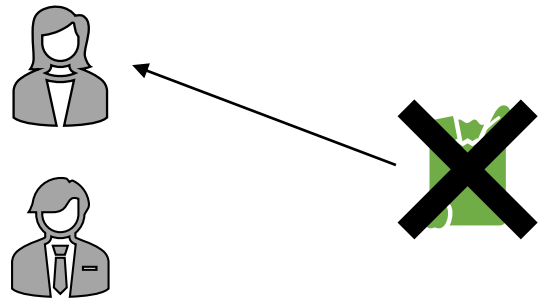
- n agents, m indivisible chores
- Agent i has disutility $d_{ij} > 0$ for chore j



- Allocation $x = (x_1, .. x_n)$ is a partition of chores to agents
- Agent i gets *disutility* $d_i(x_i)$ from *bundle* x_i
- Additive disutilities: $d_i(x_i) = \sum_{j \in x_i} d_{ij}$

Fairness Notions: EF and EF1

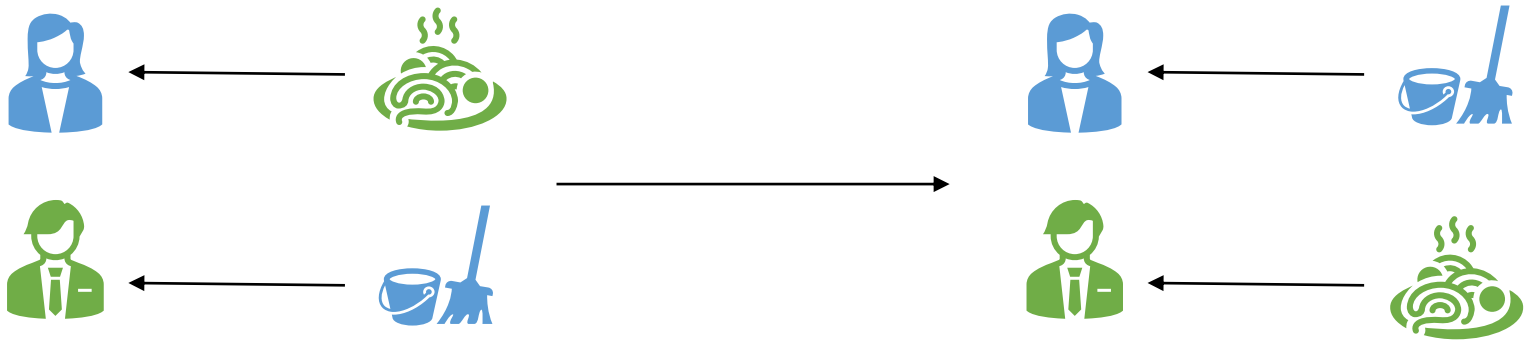
- Allocation x is *envy-free* (EF) if:
  - Every agent prefers her bundle over others
  - For all agents i, h: $d_i(x_i) \leq d_i(x_h)$
  - EF allocations need not exist



- Allocation x is *envy-free up to one chore* (EF1) if:
  - For all agents i, h, $\exists c \in x_i$ s.t. $d_i(x_i \setminus c) \leq d_i(x_h)$
  - EF1 allocations always exist
  - Envy-Cycle, Round Robin algorithms returns an EF1 allocation

Efficiency: PO

- EF1 allocations are fair but not efficient



- Allocation x is *Pareto-optimal* (PO) if there is no allocation y that makes some agent better-off and no agent worse-off

- *Can we always compute an* EF1+PO *allocation?*

Fairness + Efficiency

- Open Question: Does an EF1+PO allocation of chores exist?
- Only known results (existence and polynomial-time computation):
  - Two agents; or identical agents
  - Bivalued disutilities [G**M**Q'22, EPS'22]
    - For every $i \in N,\ j \in M$: $d_{ij} \in \{a, b\}$
  - Two types of chores [ALRS'22]

- For *goods,* EF1+PO allocations always exist
  - Maximum Nash welfare allocation $\operatorname{argmax}_x(\prod_i v_i(x_i))$ is EF1+PO [CKMPSW'17]
  - Pseudo-polynomial time algorithms which rely on *competitive equilibria* [BKM'18, G**M**'21]

EF1+PO for chores: Our results

- Open Question: Does an EF1+PO allocation of chores exist?
- Existence and polynomial-time computation known for:

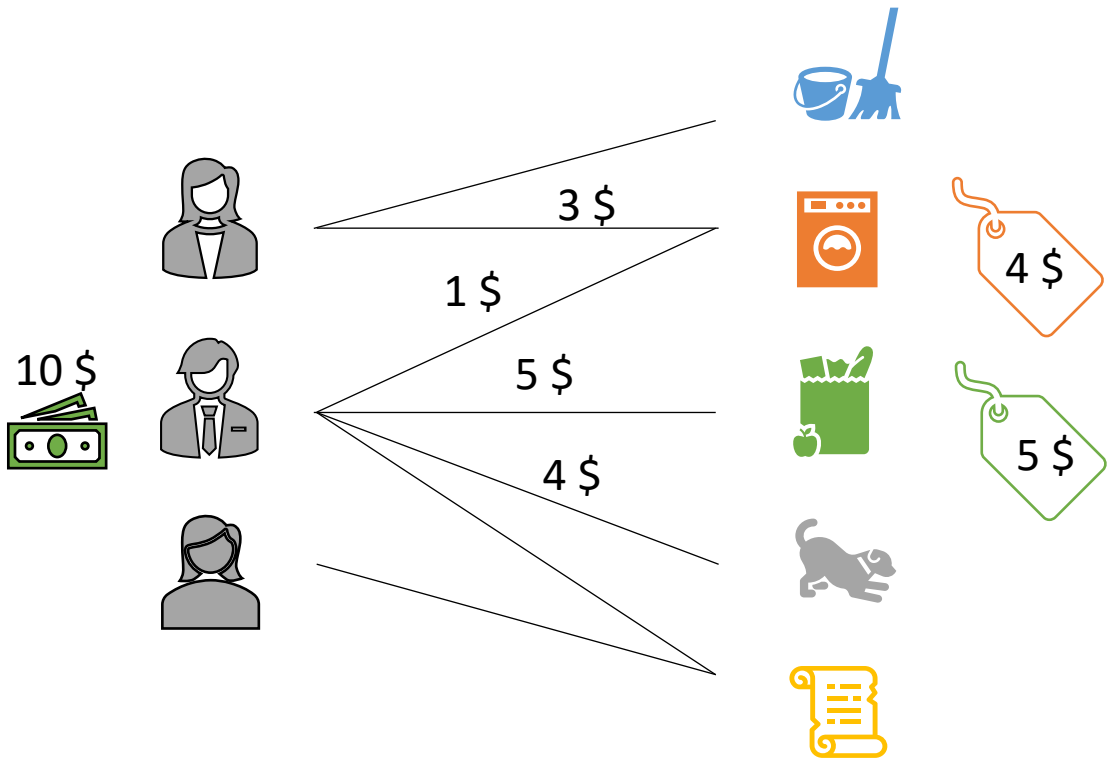
  Two agents; identical agents; bivalued chores; two types of chores

- Theorem 1: An EF1+PO allocation of chores exists and can be computed in polynomial time for $n = 3$ agents.

- Theorem 2: An EF1+PO allocation of chores exists and can be computed in polynomial time when there are two types of agents.

Two disutility functions $d_1(\cdot), d_2(\cdot)$

Competitive allocation of chores



Fisher Market for chores

- n agents, agent i with minimum salary s_i
- m chores, chore j with payment $p_j > 0$
- Chores can be fractionally allocated

Competitive equilibrium (x, p)

- All agents earn their minimum salary
- All chores are completely allocated
- Agents only perform chores that give them *minimum pain-per-buck (MPB):*

$$x_{ij} > 0 \Rightarrow \frac{d_{ij}}{p_j} = \min_c \frac{d_{ic}}{p_c}$$

Competitive equilibria for fair division

- **Second Welfare Theorem:** For a CE (x, p), the allocation x is PO
- Payment $p(x_i) = \sum_{j \in x_i} p_j$ can be used as a proxy for disutility $d_i(x_i)$
- To get an EF1+PO allocation, enough to compute an integral CE (x, p) that is payment EF1 (pEF1), i.e.:

For all i, h: $\exists c \in x_i$ s.t. $p(x_i \setminus c) \leq p(x_h)$

Algorithm overview

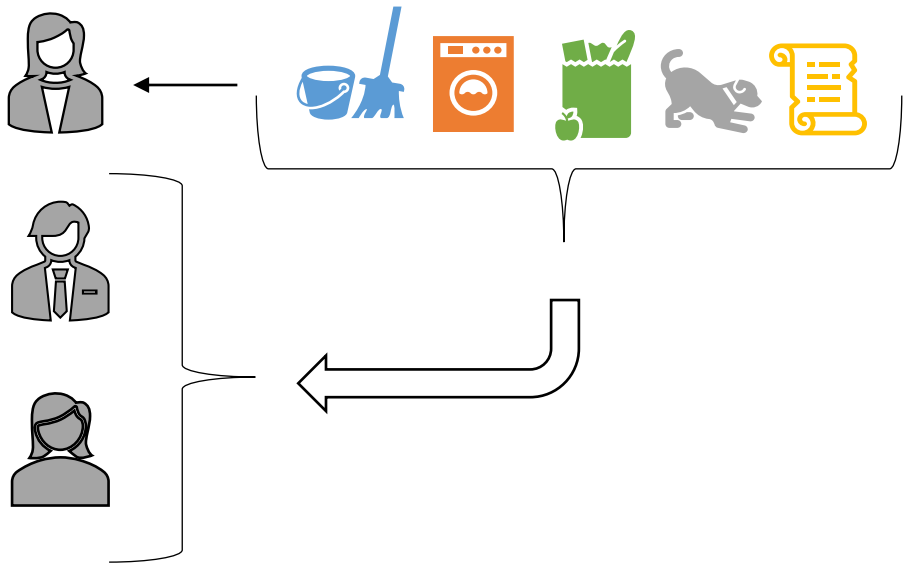
- Start with an integral CE (x, p)
- While (x, p) is not pEF1:
  - Transfer chores away from **Big Earner (BE)**: $\text{argmax}_h \min_{c \in x_h} p(x_h \setminus c)$

    and towards **Least Earner (LE)**: $\text{argmin}_i \, p(x_i)$
  - If such a transfer is not possible, raise payments of chores belonging to big earner.
  - Always maintain the minimum pain per buck (MPB) condition.

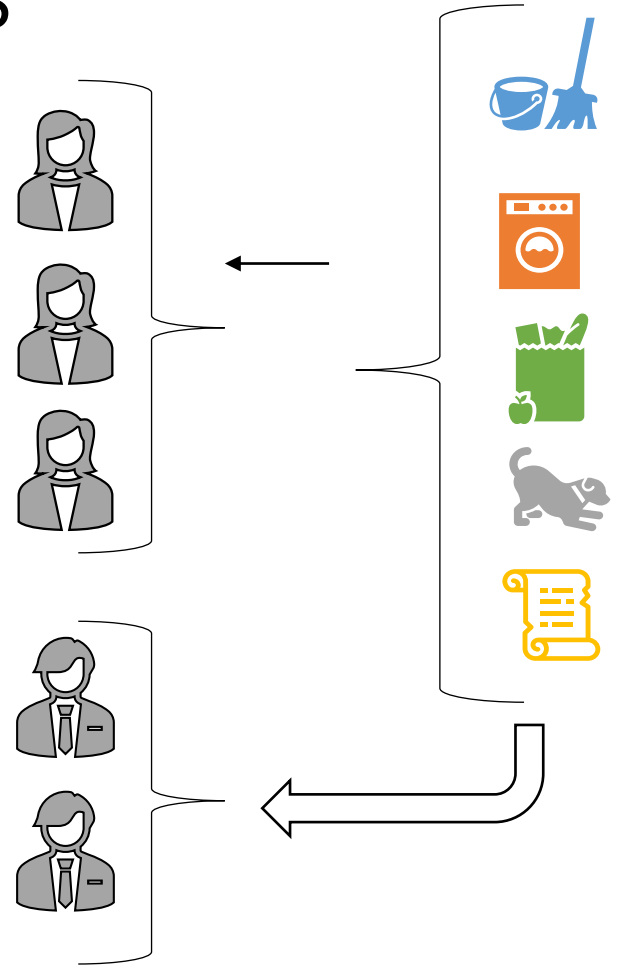
PO 

EF1+PO

Key ideas in our algorithms



$n = 3$ agents



Allocate chores to agents of same type using the Round Robin algorithm

Two types of agents

Summary

- EF1+PO allocation of chores exists and can be computed in polynomial time when there are three agents or two types of agents.
- Also show EFX+PO for three bivalued agents in polynomial time.

- For full details, please visit https://arxiv.org/abs/2212.02440

Thanks for your attention!

Email: aniket2@illinois.edu