A Simple Vision for Fair Division

Yair Zick, UMass Amherst

Fair And Efficient Allocations for Matroid Rank Functions

Fair allocation with non-additive preferences

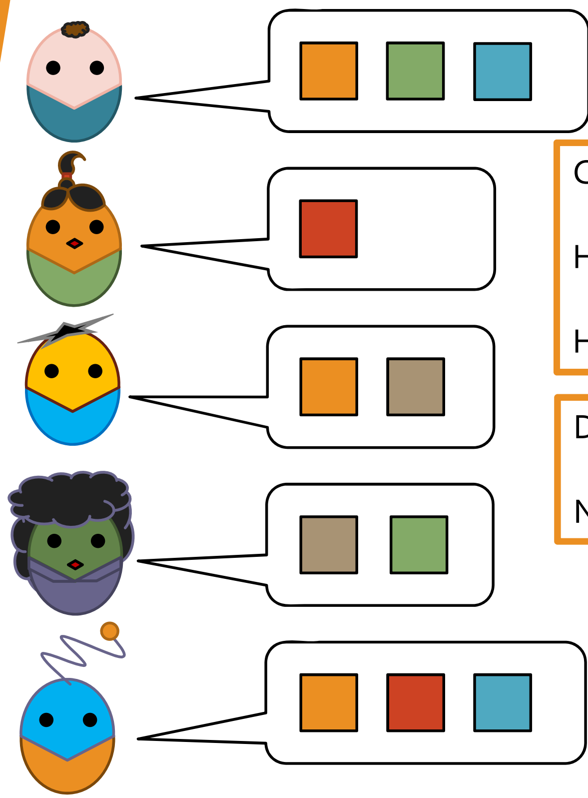
Intro to Programming

Intro to OS

Intro to AI

Intro to ML

Algo. Design



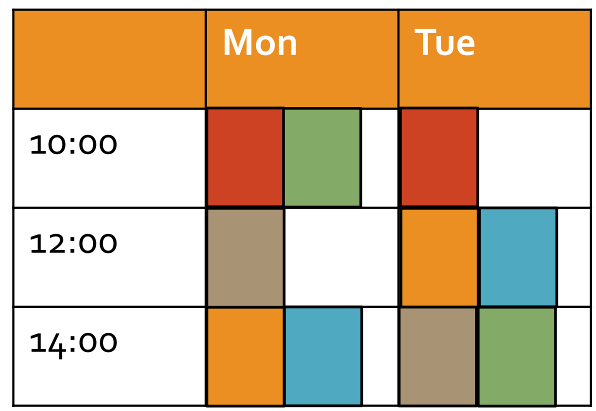
Courses scheduled at different times

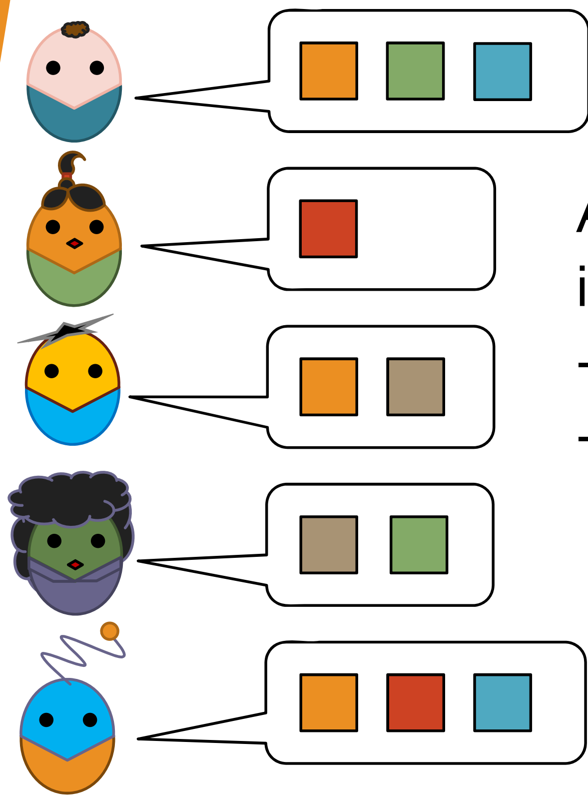
Have schedule conflicts

Have limited capacity

Different course preferences

Need to fit in schedule





Assign courses to students in a way that is

- Efficient
- Fair

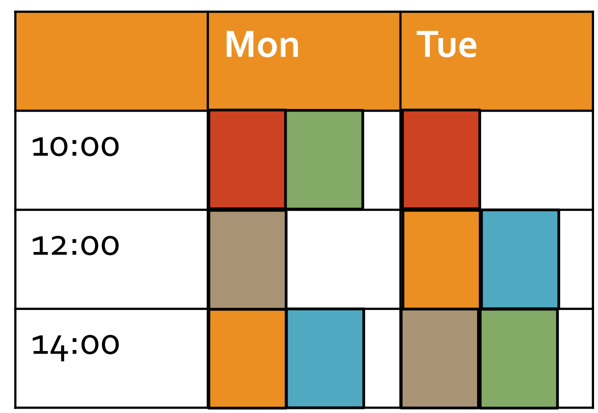
Courses scheduled at different times

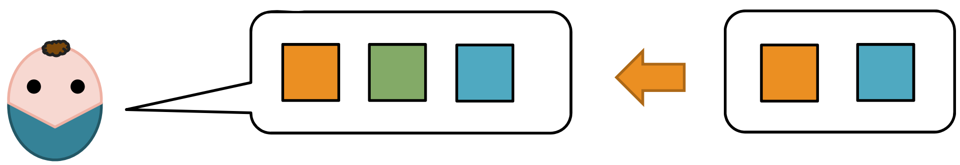
Have schedule conflicts

Have limited capacity

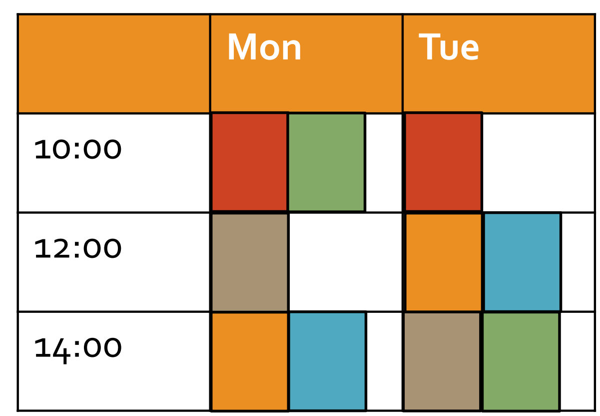
Different course preferences

Need to fit in schedule





$v_i(\square) = v_i(\square) = 1$ but $v_i(\square\square) = 1$ as well…



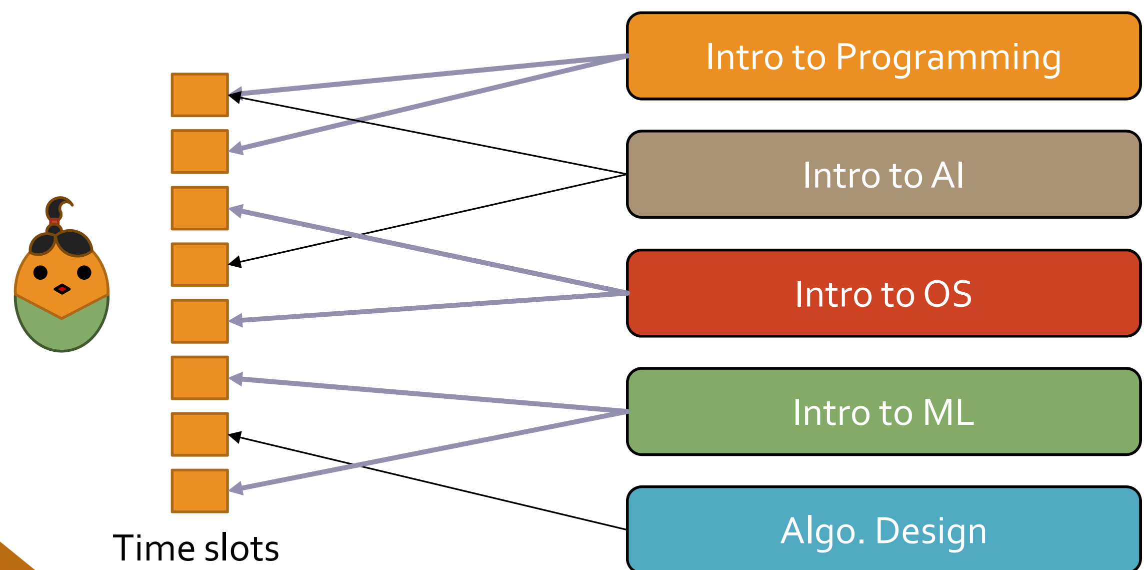
The Setup

Players $N = \{1, \ldots, n\}$ and objects $O = \{o_1, \ldots, o_m\}$

Each player $i \in N$ has a **valuation function** over **subsets of objects**:

$$v_i: 2^O \to \mathbb{R}_+$$

Matching-based valuations (OXS): $v_i(S) = |M_i(S)|$ where $M_i(S)$ is the best matching of i's approved items to i's schedule.



The Setup

Players $N = \{1, \dots, n\}$ and objects $O = \{o_1, \dots, o_m\}$

Each player $i \in N$ has a **valuation function** over **subsets of objects**:

$$v_i: 2^O \to \mathbb{R}_+$$

Matching-based valuations (OXS): $v_i(S) = |M_i(S)|$ where $M_i(S)$ is the best matching of i's approved items to i's schedule.

Submodular valuations (SUB):
Let $\Delta_i(S, o) \stackrel{\text{def}}{=} v_i(S \cup o) - v_i(S)$.
For every $i \in N$, v_i is submodular if for every $S \subseteq T \subseteq O$ and every $o \in O \setminus T$:

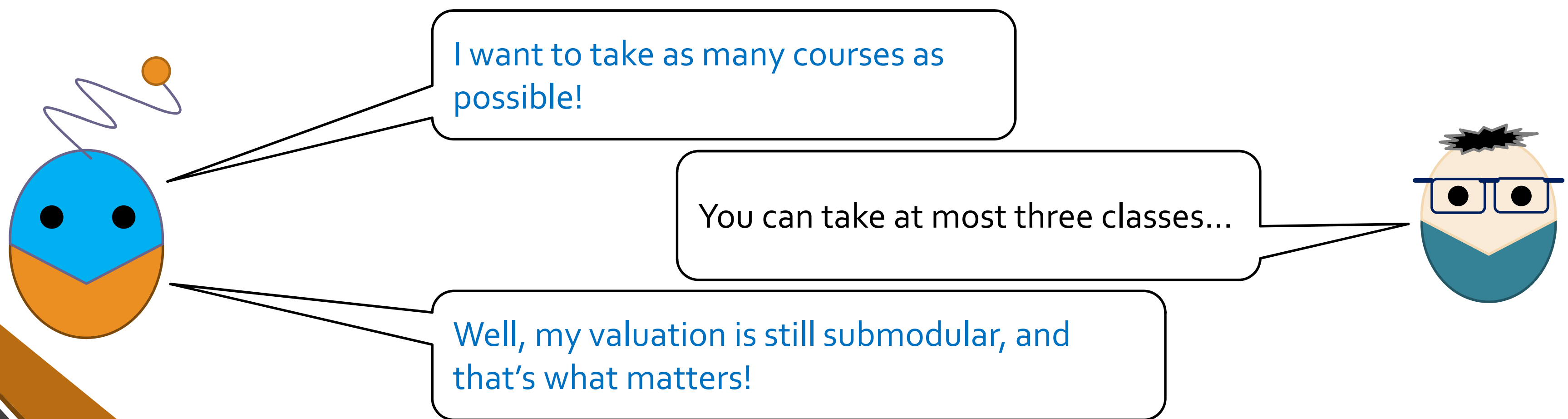
$$\Delta_i(T, o) \leq \Delta_i(S, o)$$

Adding items to 'bigger' groups has lower marginal benefit.
Binary $(0, 1)$ benefits: marginal benefit of another item is either 0 or 1.

The Setup

Submodular functions capture:

- Course allocation
- Allocating public housing to different ethnic groups
- Shift allocation
- Capacity constraints: adding **upper bounds** on # of allowed items retains submodularity.

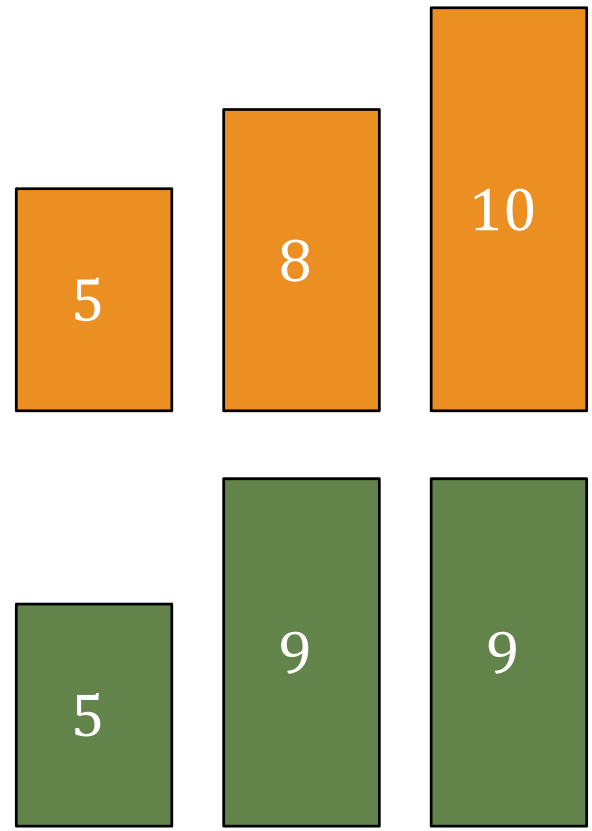


Welfare Criteria

Welfare Guarantees

- Utilitarian: $USW(A) \stackrel{\text{def}}{=} \sum_i v_i(A_i)$
- Egalitarian: $ESW(A) \stackrel{\text{def}}{=} \min_i v_i(A_i)$
- Leximin – maximize minimal welfare, then 2nd minimal welfare, then 3rd minimal welfare..
- Nash Welfare: $NSW(A) \stackrel{\text{def}}{=} \prod_i v_i(A_i)$

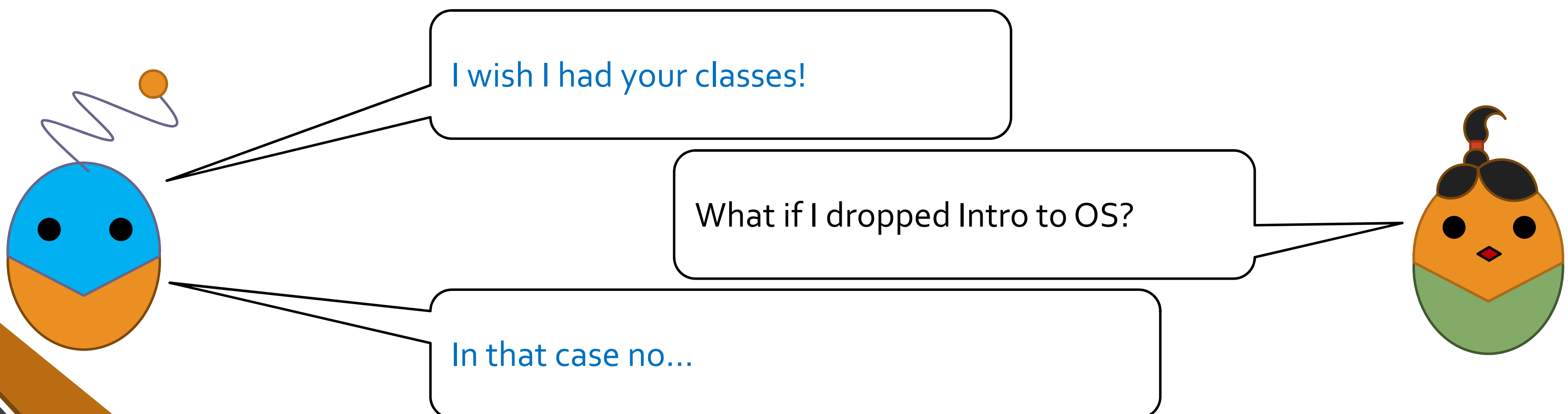
Pareto Efficiency: for any other allocation A', if there is some $j \in N$ such that $v_j(A'_j) > v_j(A_j)$ then there is some $i \in N$ such that $v_i(A'_i) < v_i(A_i)$.



Justice Criteria

Fairness Guarantees

- **Envy-Freeness (EF):** for every $i, j \in N$, $v_i(A_i) \geq v_i(A_j)$.
- **Envy-Freeness Up to One Good (EF-1):** for every $i, j \in N$, there is some $o \in A_j$ s.t. $v_i(A_i) \geq v_i(A_j \setminus o)$.



So Much Good News!

- **For $(0,1)$-SUB valuations:**
  - an EF-1 and USW-optimal allocation always exists, and can be computed in poly-time
  - Leximin and Max Nash allocations are EF-1
  - Moreover, leximin allocations are the minimizers of any symmetric strictly convex function over the utilitarian optimal allocations.
- **For $(0,1)$-OXS valuations:**
  - Leximin/Max Nash allocation can be found in poly-time.

Most results in the fair allocation literature focus on additive valuations, this is one of the first to venture out of this domain.

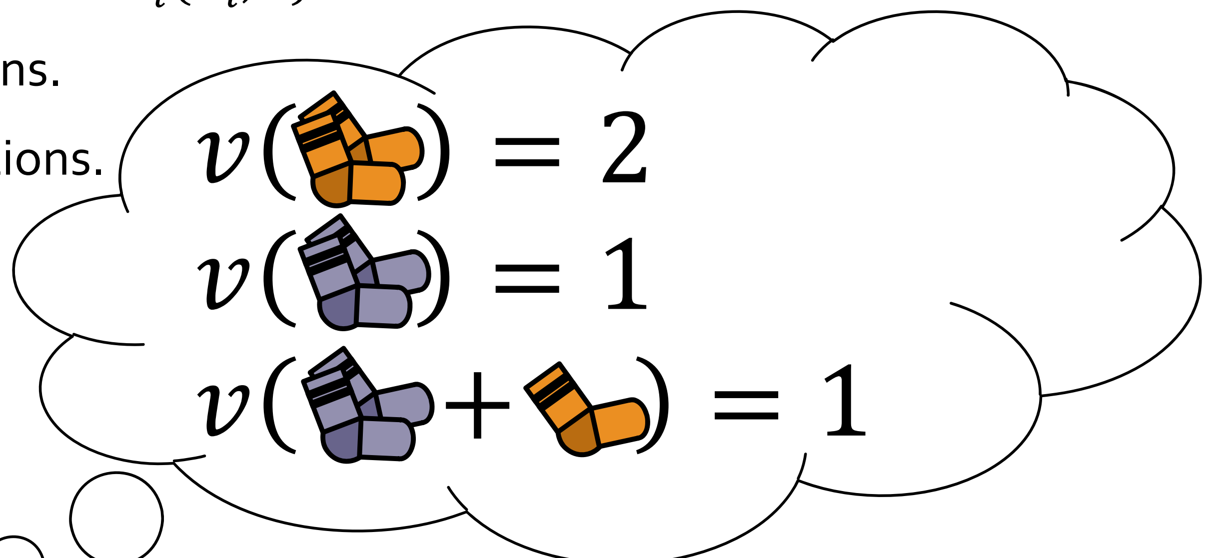
USW+EF1 Algorithm

Key Property: Transferability

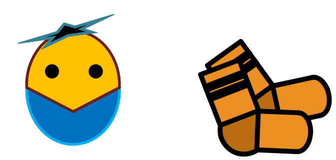
If player i envies player j, then there exists some item $o \in A_j$ such that

$$\Delta_i(A_i, o) > 0$$

- **Trivial** for additive valuations.
- **Not true** for general valuations.



Transferability holds whenever players have monotone submodular valuations!



Lemma:
Suppose that player i envies player j under allocation A and v_i is a monotone submodular valuation function, then there exists some item $o \in A_j$ such that $\Delta_i(A_i, o) > 0$.

Proof:

Suppose that for every $o \in A_j$, $\Delta_i(A_i, o) = 0$.

Let $A_j = \{o_1, \ldots, o_r\}$ and let $A_j^t \stackrel{\text{def}}{=} \{o_1, \ldots, o_t\}$ (with $A_j^0 = \emptyset$). Then

$$v_i(A_i \cup A_j) - v_i(A_i) = \sum_t \Delta_i\left(A_i \cup A_j^{t-1}, o_t\right)$$

By submodularity: $\Delta_i\left(A_i \cup A_j^{t-1}, o_t\right) \leq \Delta_i(A_i, o_t) = 0$

Therefore, $v_i\left(A_i \cup A_j\right) - v_i(A_i) = 0$

But… $v_i\left(A_j\right) \leq v_i\left(A_i \cup A_j\right) = v_i(A_i) < v_i\left(A_j\right)$

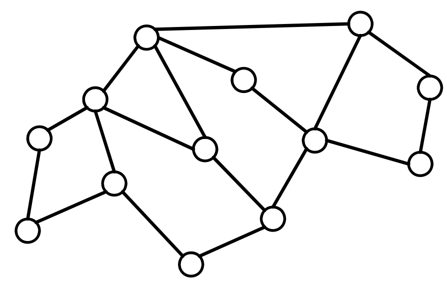
Matroids and Binary Submodular Valuations

- Independent sets of vectors in $\mathbb{R}^n$

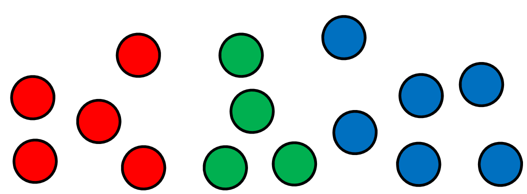
$(1,0,0), (0,1,0), (0,0,1)$

$(0,1,1), (1,1,0)$

- Forests in a graph



- Set Partitions



"You may take at most three red items, at most one green item, and at most two blue items"

Matroids

A **matroid** is given by a ground set of elements E, and a family $\mathcal{I}$ of independent subsets of E satisfying:

I1) $\emptyset \in \mathcal{I}$

I2) If $Y \in \mathcal{I}$ and $X \subseteq Y$, then $X \in \mathcal{I}$

I3) If $X, Y \in \mathcal{I}$ and $|X| < |Y|$ then there exists some $e \in Y$ such that $X \cup e \in \mathcal{I}$

USW+EF1 Algorithm

Key Concept: Matroids:

Given a set of elements E, let $\mathcal{I}$ be a family of subsets of E satisfying

I1) $\emptyset \in \mathcal{I}$

I2) If $Y \in \mathcal{I}$ and $X \subseteq Y$, then $X \in \mathcal{I}$

I3) If $X, Y \in \mathcal{I}$ and $|X| < |Y|$ then there exists some $e \in Y$ such that $X \cup e \in \mathcal{I}$

Matroids and Submodularity:

Given a matroid $\langle E, \mathcal{I} \rangle$, let $r: 2^E \to \mathbb{R}$ be $r(S) = \max\{|X|: X \subseteq S, X \in \mathcal{I}\}$

The rank function r is $(0,1)$-SUB, but there's more!

Every $(\mathbf{0}, \mathbf{1})$-SUB function is the rank function of some matroid.

Matroids and Optimization

Greedy Techniques Work:

Given a matroid $\langle E, \mathcal{I} \rangle$ and a weight function $w: E \to \mathbb{R}$, it is possible to find an independent set I maximizing $\sum_{e \in I} w(e)$ using a simple greedy algorithm (e.g. Kruskal's algorithm for min spanning tree).

Union of Matroids is a Matroid: given matroids $\langle E, \mathcal{I}_1 \rangle$ and $\langle E, \mathcal{I}_2 \rangle$, $\langle E, \mathcal{I}_1 \cup \mathcal{I}_2 \rangle$ is a matroid!

Intersection of Matroids: is not necessarily a matroid, but can still find an independent set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ in polynomial time.

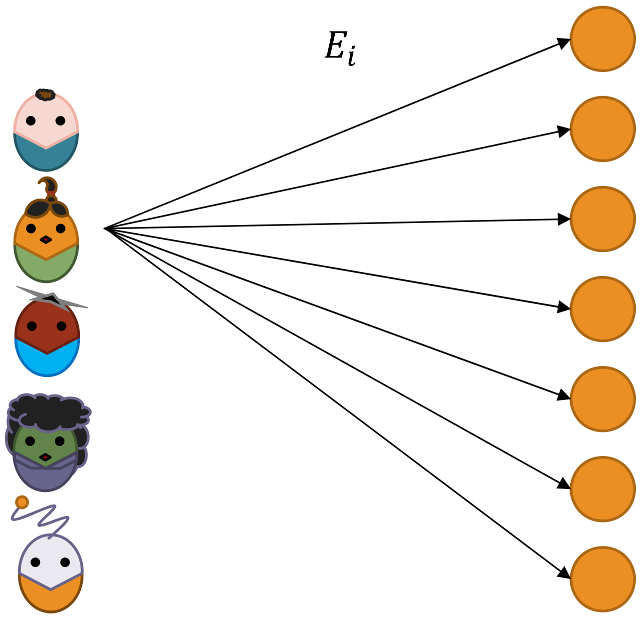
USW+EF1 Algorithm

The algorithm:

1. Compute a USW optimal allocation in poly-time (using matroids)
2. Transfer items between agents until all (non EF-1) envy is eliminated (using transferability)

This algorithm will work for any valuation class where (a) transfers are possible and (b) we can bound the welfare loss due to transfers

Computing a USW Optimal Allocation



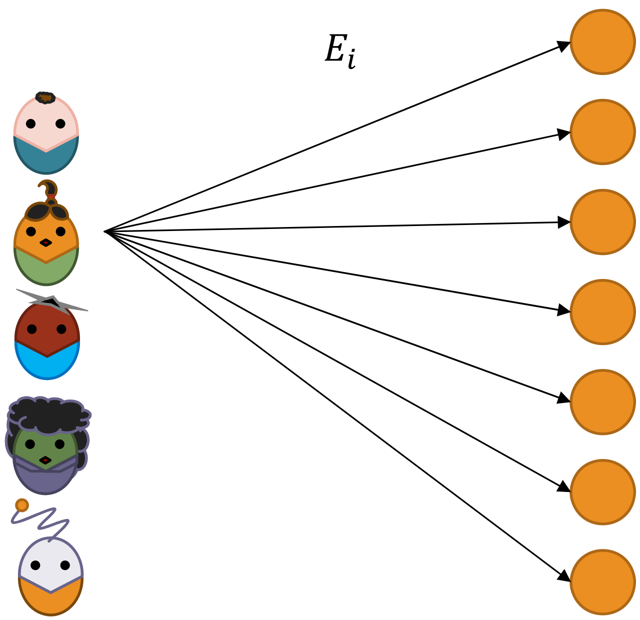
Let $E_i = \{\{i, o\}: o \in O\}$. Let $E = \bigcup_i E_i$. Given $X \subseteq E$, let

$$r_i(X) = v_i(\{o \in O: \{i, o\} \in X \cap E_i\})$$

r_i is a $(0,1)$-SUB function over E and thus induces a matroid $\mathcal{I}_i$.

The union of matroids is a matroid: $\mathcal{I} = \bigcup_i \mathcal{I}_i$ is a matroid.

Computing a USW Optimal Allocation



Let $E_i = \{\{i, o\}: o \in O\}$. Let $E = \bigcup_i E_i$. Given $X \subseteq E$, let

$$r_i(X) = v_i(\{o \in O: \{i, o\} \in X \cap E_i\})$$

r_i is a $(0,1)$-SUB function over E and thus induces a matroid $\mathcal{I}_i$.

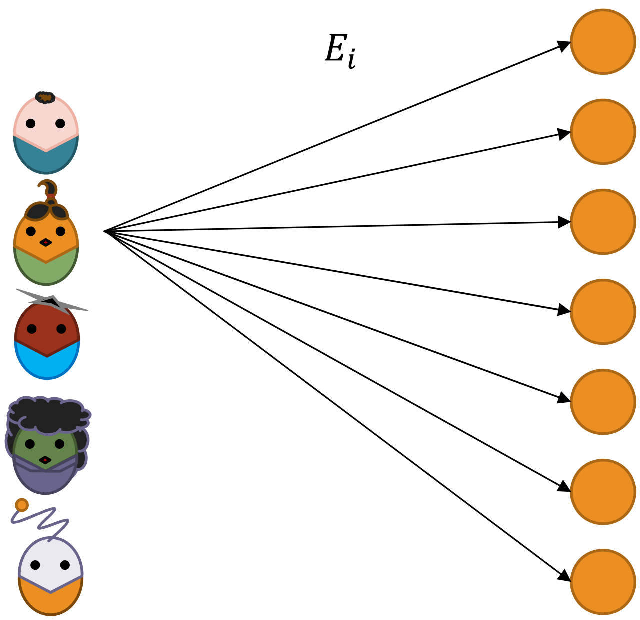
The union of matroids is a matroid: $\mathcal{I} = \bigcup_i \mathcal{I}_i$ is a matroid.

Need to also ensure that each item is allocated to exactly one player! This is a **partition matroid**:

$$\mathcal{O} = \{X \subseteq E: |X \cap E_o| \leq 1\}$$

Where $E_o = \{\{i, o\}: i \in N\}$

Computing a USW Optimal Allocation



Let $E_i = \{\{i,o\}: o \in O\}$. Let $E = \bigcup_i E_i$. Given $X \subseteq E$, let

$$r_i(X) = v_i(\{o \in O: \{i,o\} \in X \cap E_i\})$$

$\mathcal{I}$ corresponds to players' valuations on items

$\mathcal{O}$ enforces that every item goes to at most one player

Elements in $\mathcal{I} \cap \mathcal{O}$ correspond to clean, valid allocations.

A maximal cardinality allocation in $\mathcal{I} \cap \mathcal{O}$ is one that maximizes USW.

Finding the largest common independent set in two matroids can be done in $\mathcal{O}\left(|E|^3\theta\right)$ (θ – time to test independence). Thus poly time!

Envy-Induced Transfers

We found a USW maximizing allocation A using the magic of matroids, hurray!

Could still not be EF-1 though.

Let us define $\Phi(A) = \sum_i v_i(A_i)^2$: a **potential function**.

Suppose that player i envies player j for more than one item.

Then player j can transfer an item to player i that will strictly benefit player i, and maybe hurt player j by at most 1.

Each such transfer (strictly) reduces the value of Φ.

The maximal value of Φ is $\leq \left(\sum_i v_i(A_i)\right)^2 \leq m^2$, so at most m^2 such transfers will occur.

Let us define $\Phi(A) = \sum_i v_i(A_i)^2$: a **potential function**.

Suppose that player i envies player j for more than one item.

Then player j can transfer an item to player i that will strictly benefit player i, and maybe hurt player j by at most 1.

Let A be the allocation before the transfer and A' be the allocation after the transfer. We assume that the allocation is **clean**. Then:

$$\Phi(A') - \Phi(A) = \sum_k v_k(A'_k)^2 - \sum_k v_k(A_k)^2$$
$$= v_i(A'_i)^2 + v_j\left(A'_j\right)^2 - v_i(A_i)^2 - v_j\left(A_j\right)^2$$
$$= (v_i(A_i) + 1)^2 + \left(v_j\left(A_j\right) - 1\right)^2 - v_i(A_i)^2 - v_j\left(A_j\right)^2$$
$$= 2\left(1 + v_i(A_i) - v_j\left(A_j\right)\right)$$

Let us define $\Phi(A) = \sum_i v_i(A_i)^2$: a **potential function**.

Suppose that player i envies player j for more than one item.

Then player j can transfer an item to player i that will strictly benefit player i, and maybe hurt player j by at most 1.

Let A be the allocation before the transfer and A' be the allocation after the transfer. We assume that the allocation is **clean**. Then:

$$\Phi(A') - \Phi(A) = 2\left(1 + v_i(A_i) - v_j(A_j)\right)$$

We want to show that $1 + v_i(A_i) - v_j(A_j) < 0$, or that

$$1 + v_i(A_i) < v_j(A_j)$$

If we do – we're done!

Let us define $\Phi(A) = \sum_i v_i(A_i)^2$: a **potential function**.

Suppose that player i envies player j for more than one item.

Then player j can transfer an item to player i that will strictly benefit player i, and maybe hurt player j by at most 1.

We want to show that $1 + v_i(A_i) - v_j(A_j) < 0$, or that

$$1 + v_i(A_i) < v_j(A_j)$$

If we do – we're done! We haven't used the fact that i envies j for more than one item yet, maybe that would be useful…

Technical facts about binary submodular functions:

1. If S is clean, then $v_i(S) = |S|$
2. For every bundle S, $v_i(S) \leq |S|$

Since i envies j for more than one item, then for every item $o \in A_j$,

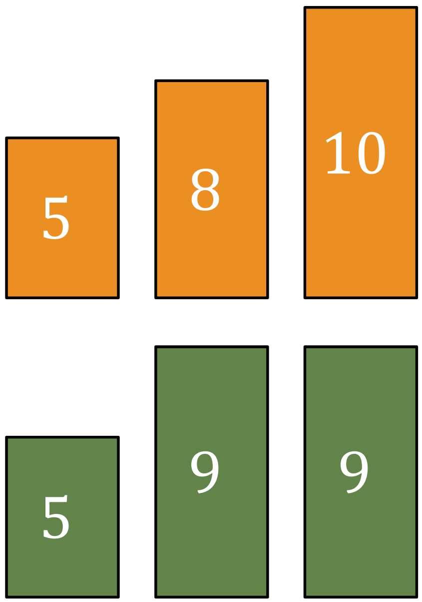
$$v_i(A_i) < v_i\left(A_j \setminus \{o\}\right) \leq \left|A_j \setminus \{o\}\right| = \left|A_j\right| - 1$$

Since A_j is clean, $v_j\left(A_j\right) = \left|A_j\right|$, and we have

$$v_i(A_i) < \left|A_j\right| - 1 = v_j\left(A_j\right) - 1 \Rightarrow v_j\left(A_j\right) > v_i(A_i) + 1$$

Leximin and Socially Optimal Allocations

Leximin allocation: maximize the unhappiest agent's welfare; then subject to that, maximize the second unhappiest agent's welfare; then subject to that…



Leximin and Socially Optimal Allocations

Utilitarian optimal does not necessarily imply leximin.

For $(0,1)$-SUB valuations:

Leximin/MNW → Pareto-Optimality → Utilitarian Social Welfare

Leximin and Socially Optimal Allocations

Lemma: let A be a socially optimal allocation that is not leximin optimal. Then there exists some pair of agents i and j and a **socially optimal** allocation A' such that

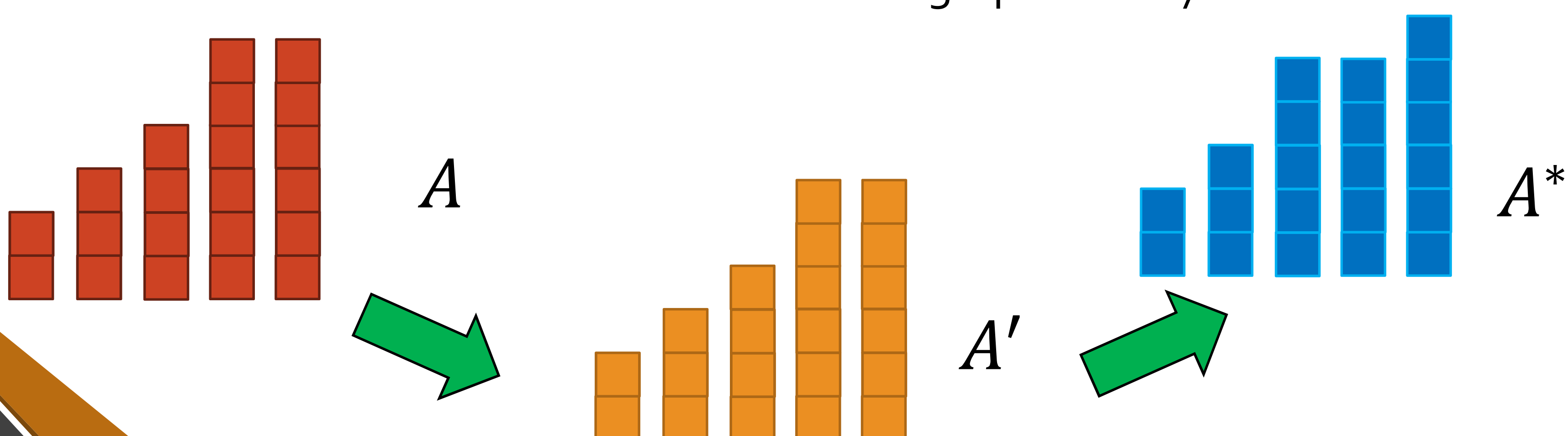
- Player i improves its welfare by 1
- Player j decreases its welfare by 1
- A' lexicographically dominates A.

Proof idea: find an allocation A' that minimizes the total symmetric difference between the leximin-optimal allocation A^* and allocations that have the same welfare vector as A. A pair i, j is guaranteed to exist that will bring us closer to A^* - the leximin optimal allocation.

Leximin and Socially Optimal Allocations

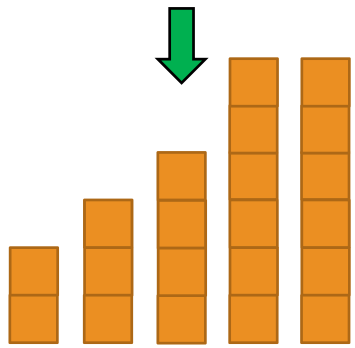
Let A be an arbitrary utilitarian optimal allocation that is not leximin, and let A^* be a leximin allocation. W.l.o.g both are clean allocations.

Take a clean allocation A' that minimizes the symmetric difference $\sum_i |A'_i \Delta A^*_i|$ over all clean allocations with the **same** lexicographic utility vector as A.

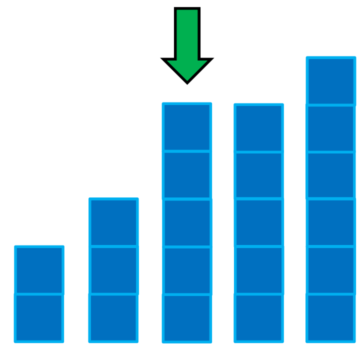


Leximin and Socially Optimal Allocations

A'

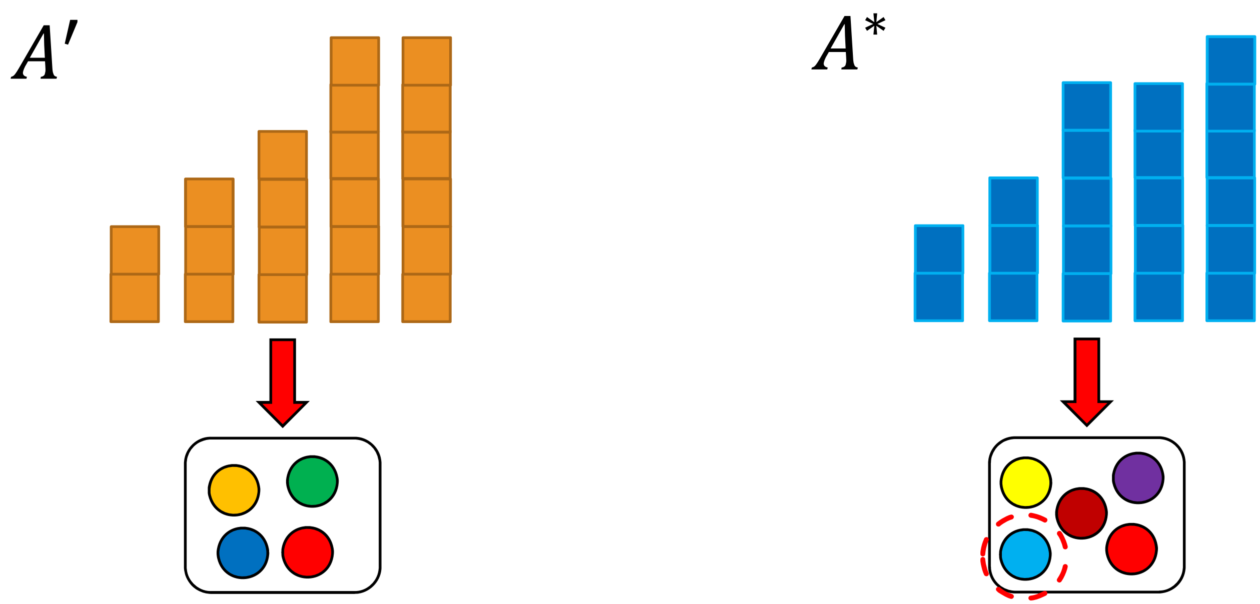


A^*



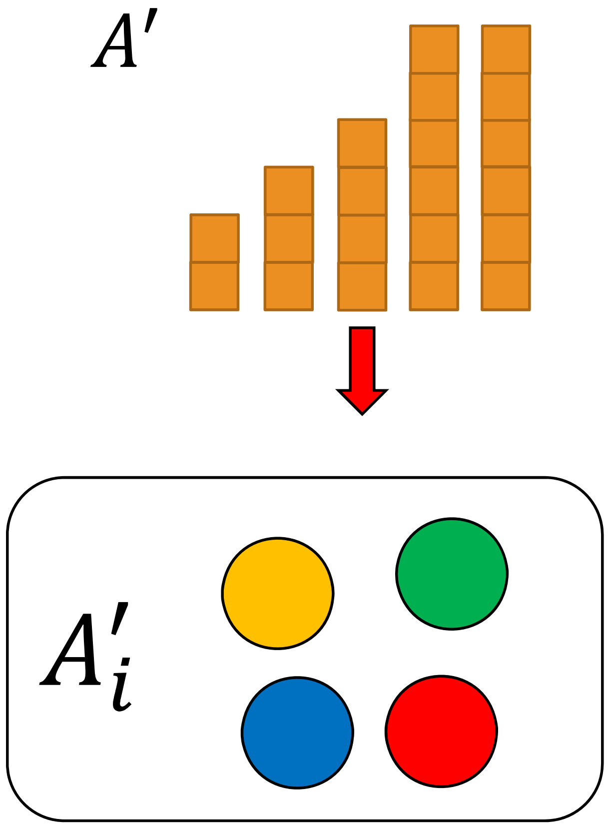
Since A^* lexicographically dominates A', there's some minimal index k such that the k-th unhappiest agent under A' is strictly worse off than the k-th unhappiest agent under A^*.

Leximin and Socially Optimal Allocations

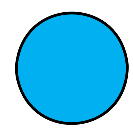


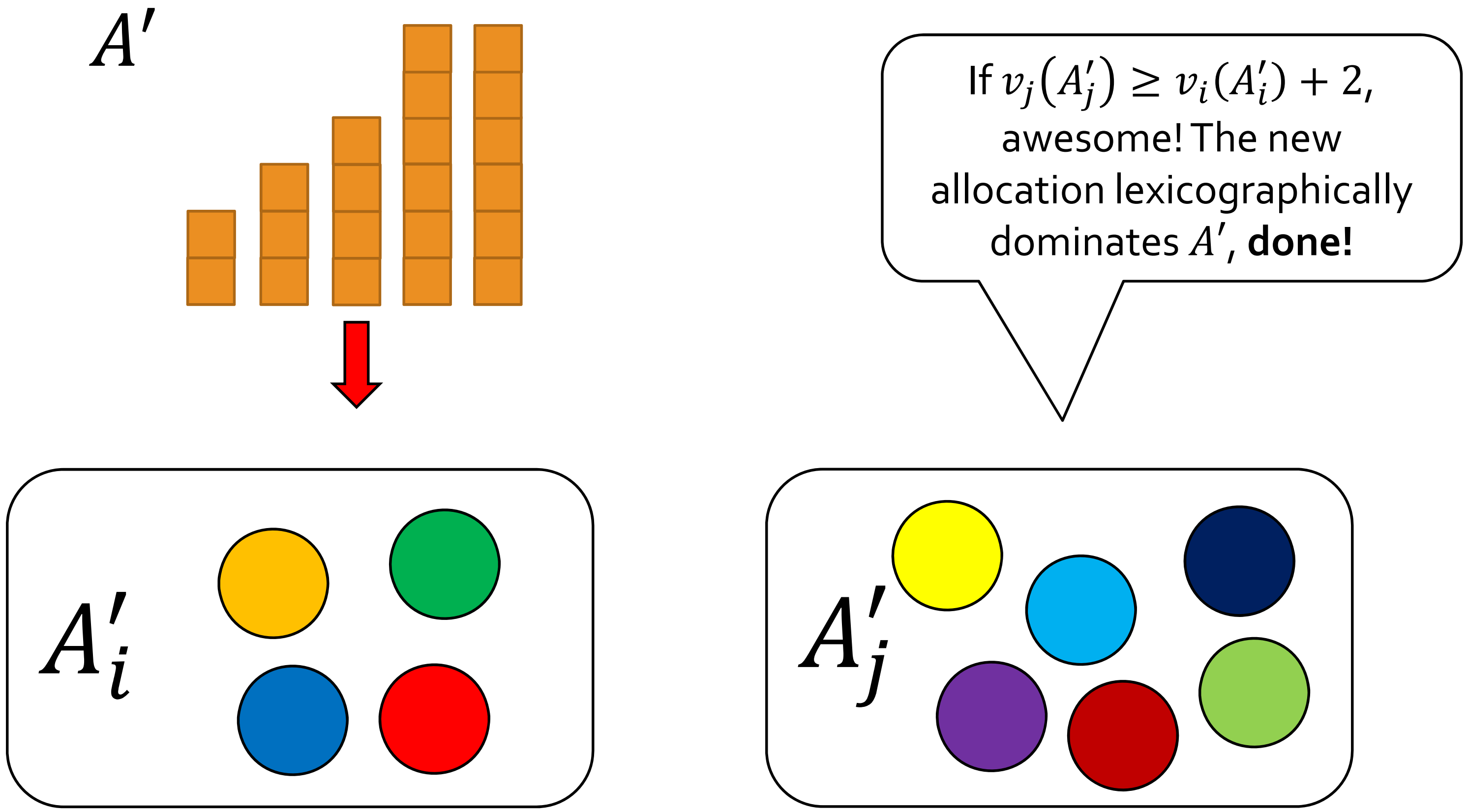
Since both allocations are clean and $|A_i^*| = v_i(A_i^*) > v_i(A_i') = |A_i'|$

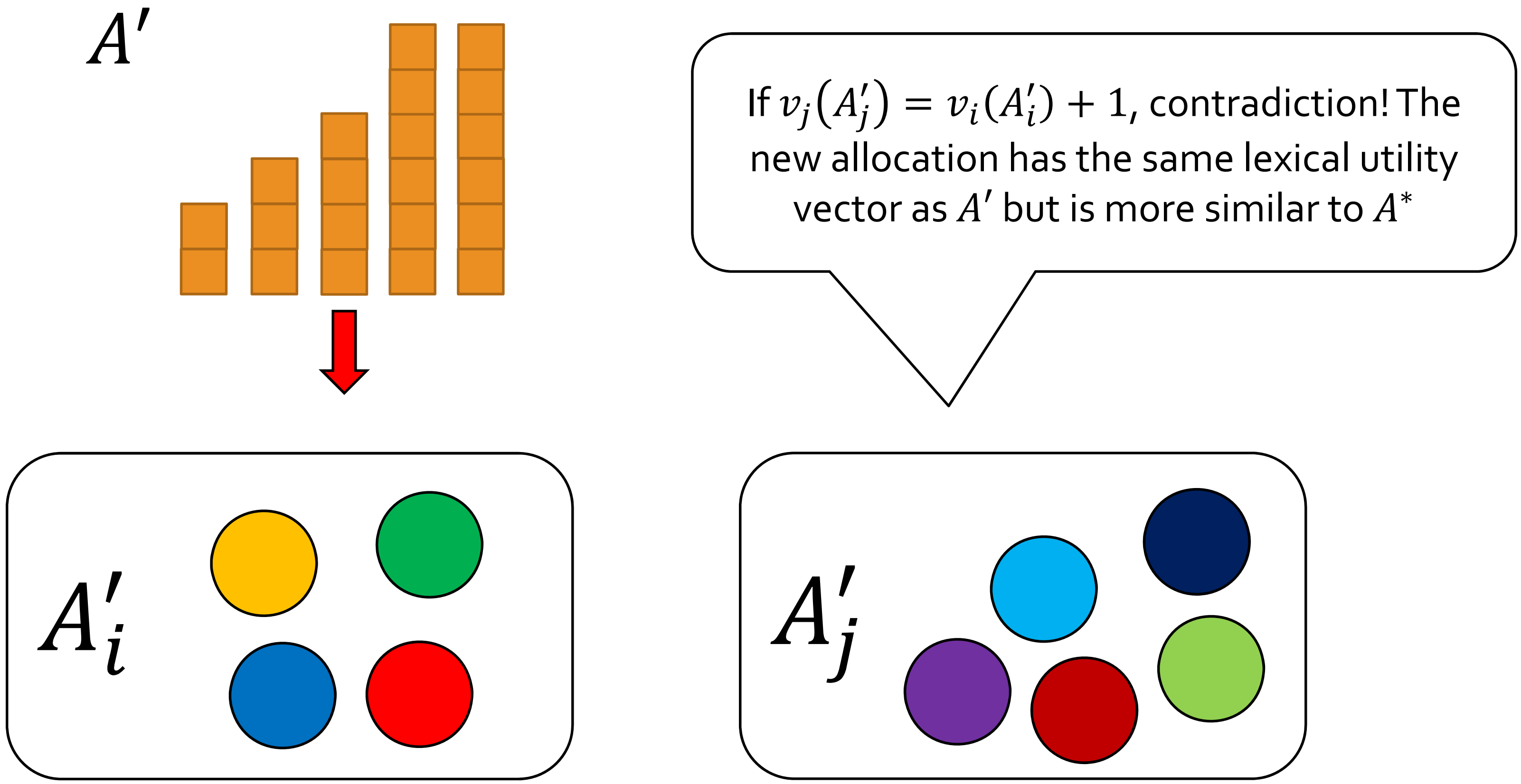
By the matroid exchange property, there is some item $o \in A_i^* \setminus A_i'$ such that $\Delta_i(A_i', o) = 1$.

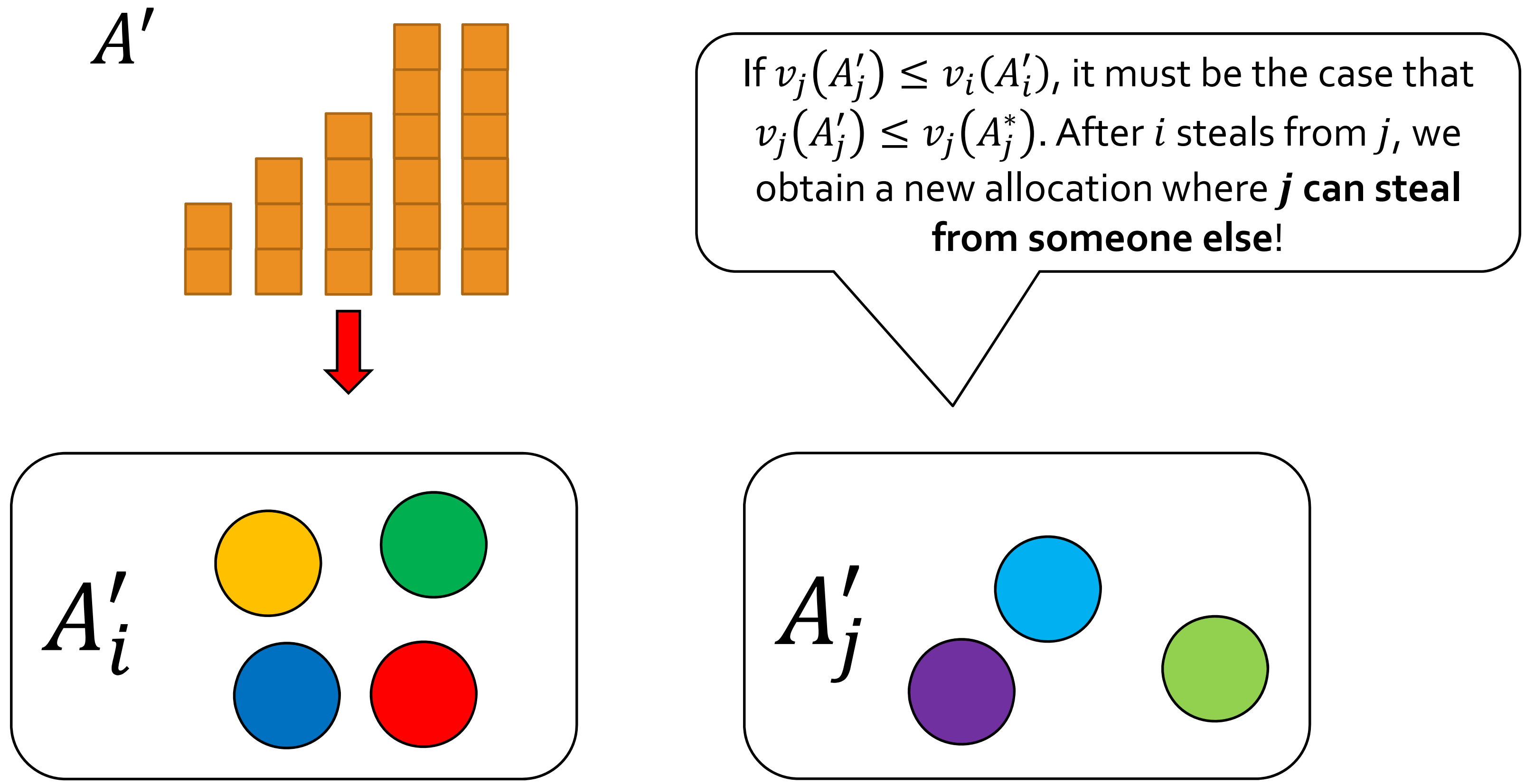


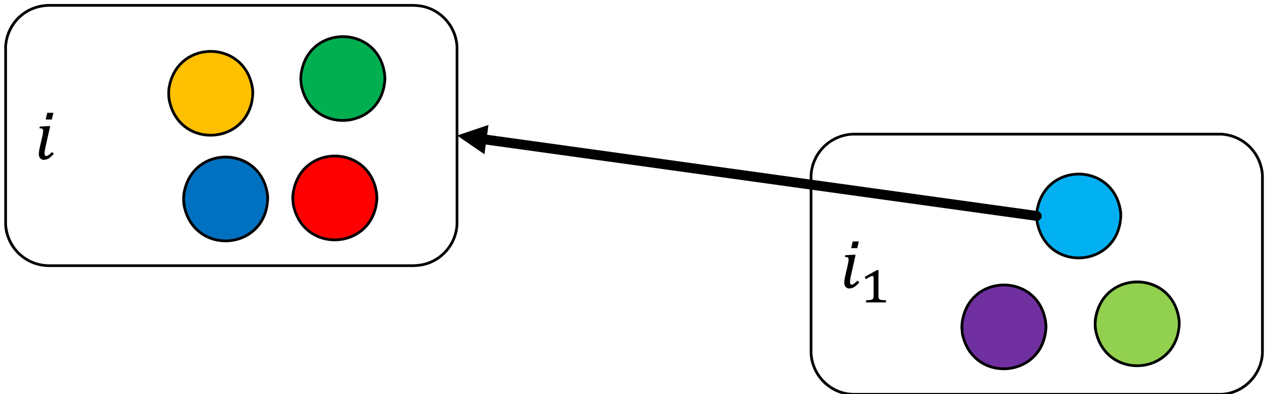
This item benefits i, and it belongs to **someone else** under A'. Call that agent j.

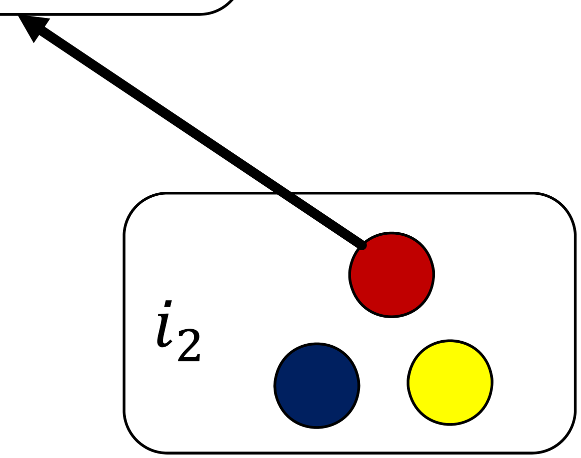



A'
A_i'
A_j'
If $v_j(A_j') \geq v_i(A_i') + 2$, awesome! The new allocation lexicographically dominates A', **done!**


A'
If $v_j(A'_j) = v_i(A'_i) + 1$, contradiction! The new allocation has the same lexical utility vector as A' but is more similar to A^*
A'_i
A'_j

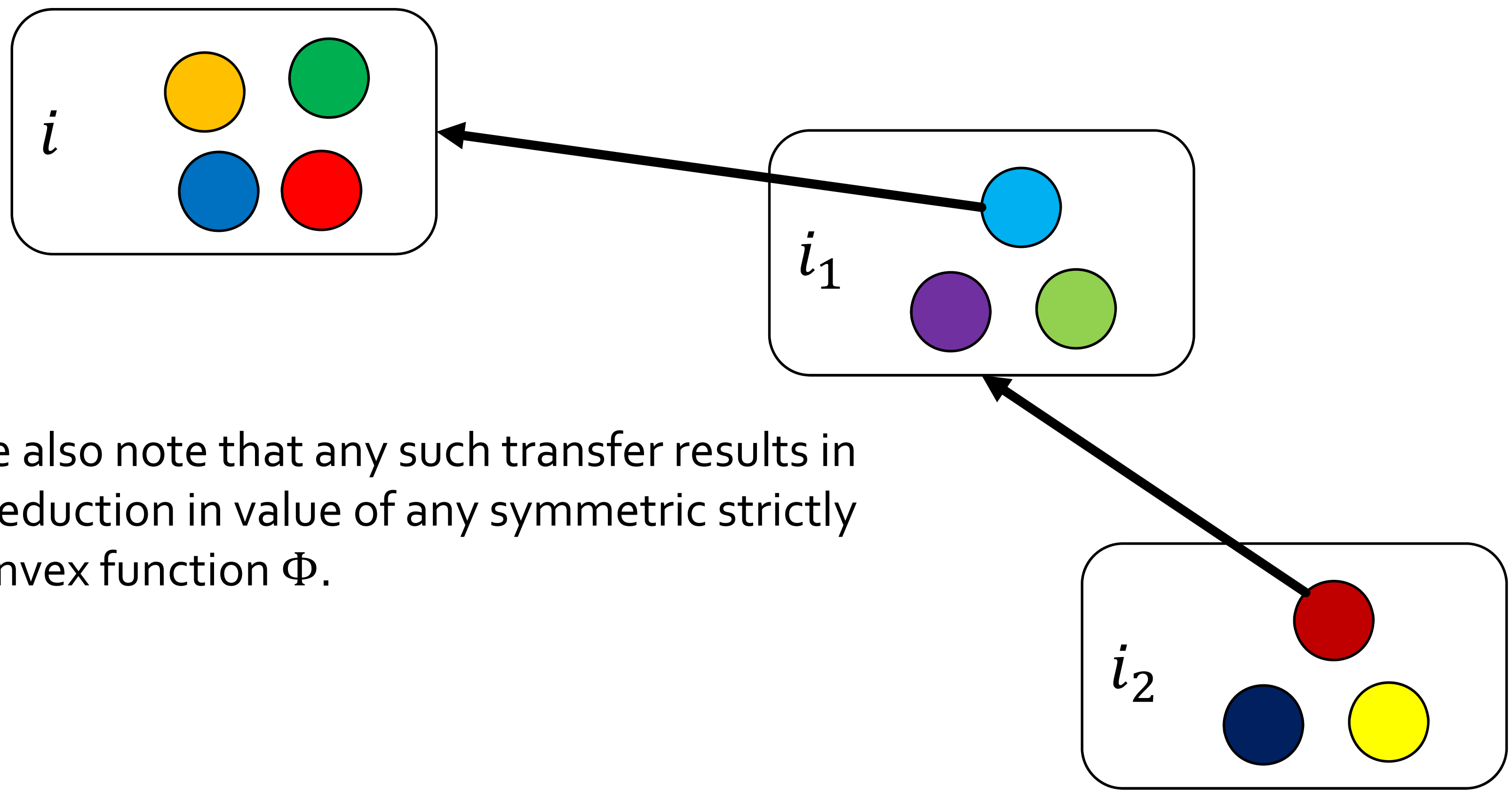

A'
A'_i
A'_j
If $v_j(A'_j) \leq v_i(A'_i)$, it must be the case that $v_j(A'_j) \leq v_j(A^*_j)$. After i steals from j, we obtain a new allocation where ***j* can steal from someone else**!





If at any point along the theft path we get a cycle, then by acting on it we get an allocation that's more like A^*, a contradiction!

Thus, it must terminate in some agent with utility ≥ 2 more than i!



We also note that any such transfer results in a reduction in value of any symmetric strictly convex function Φ.

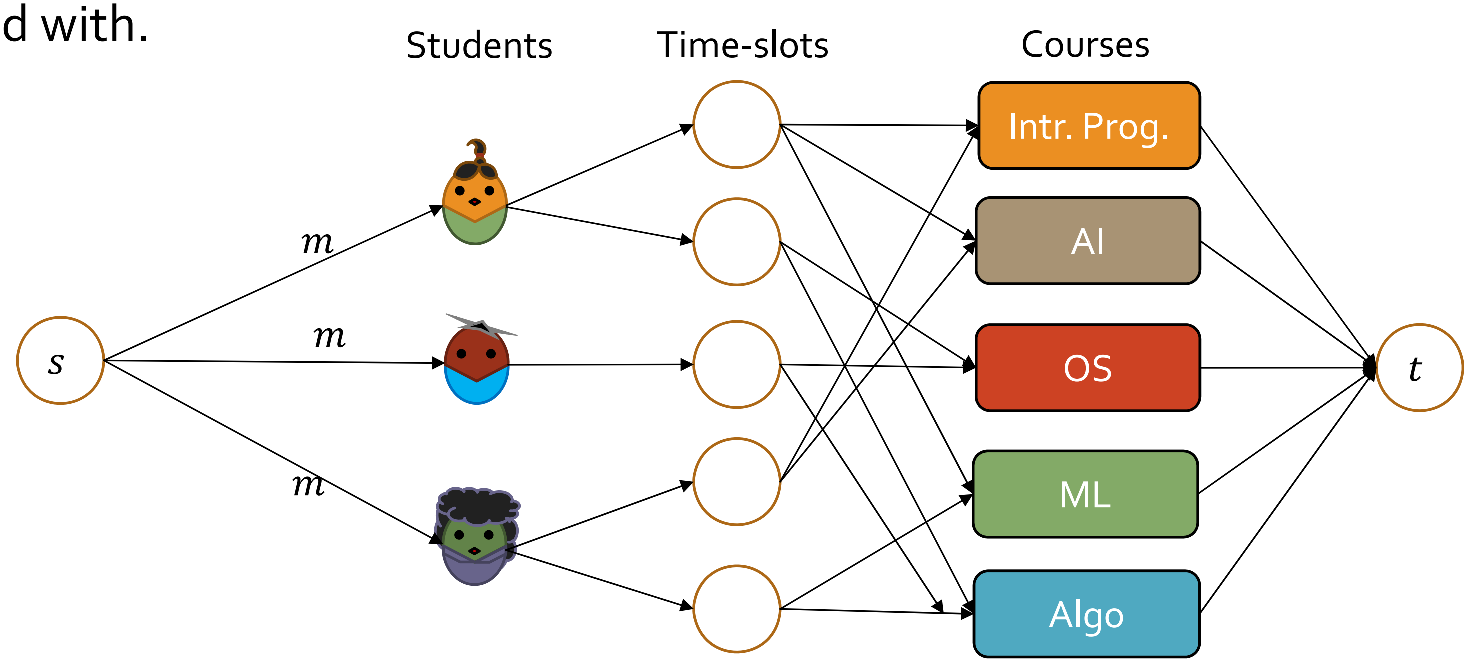
Theorem [Benabbou, Chakraborty, Igarashi and Zick, TEAC 2021]:
the following are equivalent when agents have MRF valuations

1. The allocation A^* minimizes some symmetric strictly convex function Φ over all utilitarian optimal allocations.
2. The allocation A^* maximizes some symmetric strictly concave function Ψ over all utilitarian optimal allocations.
3. The allocation A^* is leximin.
4. The allocation A^* maximizes Nash welfare.

(0,1)-OXS Valuations

Theorem: leximin/MNW allocations can be found in poly-time for (0,1)-OXS valuations

Proof Idea: construct a flow graph that optimally “flows” agents to items they’d like to be matched with.



Leximin and Socially Optimal Allocations

Theorem [Babaioff et al. 2021]: if agents have MRF valuations, then a **truthful** leximin allocation can be computed in polynomial time.

Proof Idea: further exploitation of matroid structure to find Lorenz Dominating Allocations.

Leximin and Socially Optimal Allocations

Theorem [Babaioff et al. 2021]: if agents have MRF valuations, then a **truthful** leximin allocation can be computed in polynomial time.

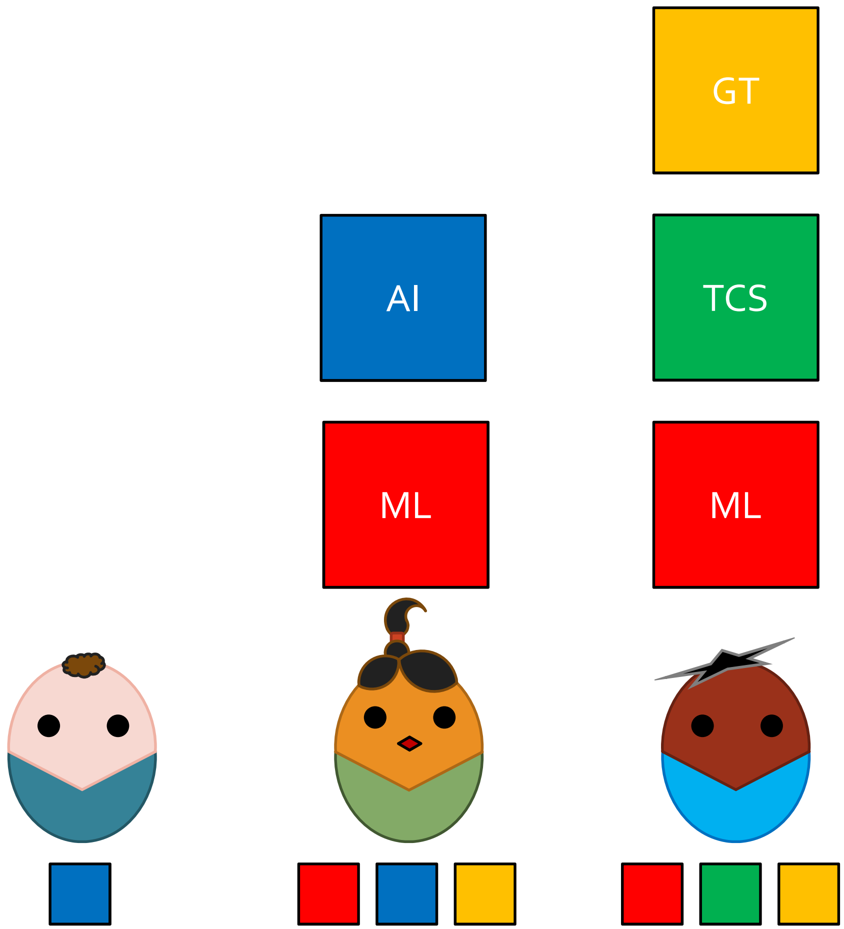
Proof Idea: further exploitation of matroid structure to find Lorenz Dominating Allocations.

These are Awfully Complicated…

Theorem [Viswanathan and Zick, 2022]: Babaioff et al.'s algorithm runs in $\mathcal{O}\left(n^6 m^{\frac{7}{2}}(\mathrm{m}+\tau)\right)$ time, where τ is the time it takes to evaluate $v_i(S)$.

Can we do better?

EF-1 Transfers Are Not Enough



Yankee Swap

```
Set $A_j \leftarrow \emptyset$ for all $j \in N$ and $A_0 = G$ be the
unassigned items.
For every agent $j \in N$, set $\text{flag}_j = \text{false}$
while $\text{flag}_j = \text{false}$ for some $j \in N$ do:
   Let $T \leftarrow \{j \in N: \text{flag}_j = \text{false}\}$
   Let $T' \leftarrow \arg\min\{v_j(A_j): j \in T\}$
   Pick an agent $i \in T'$ with highest priority
   if there is some transfer path from $i$ to $A_0$:
      execute transfer path
   else
      $\text{flag}_i = \text{true}$
return $(A_1, \dots, A_n)$
```

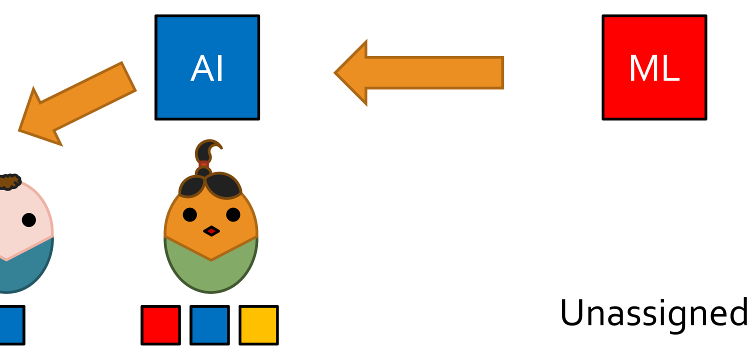
Theorem [Viswanathan and Zick, 2022]: Yankee Swap outputs a Lorenz dominating allocation, it is:

1. USW optimal
2. NSW optimal
3. EF-1
4. Leximin
5. Truthful
6. $\frac{1}{2}$-MMS

Its runtime is $\mathcal{O}\left(m^2(n+\tau)(m+n)\right)$

Key Idea 1: balancedness
When agent i's flag is set to true, any agent j that it can steal an item from has a bundle of size $\leq |A_i| + 1$

We would not permit j to get an item that would make it get such a large bundle, but rather pick i to steal an item at that round.

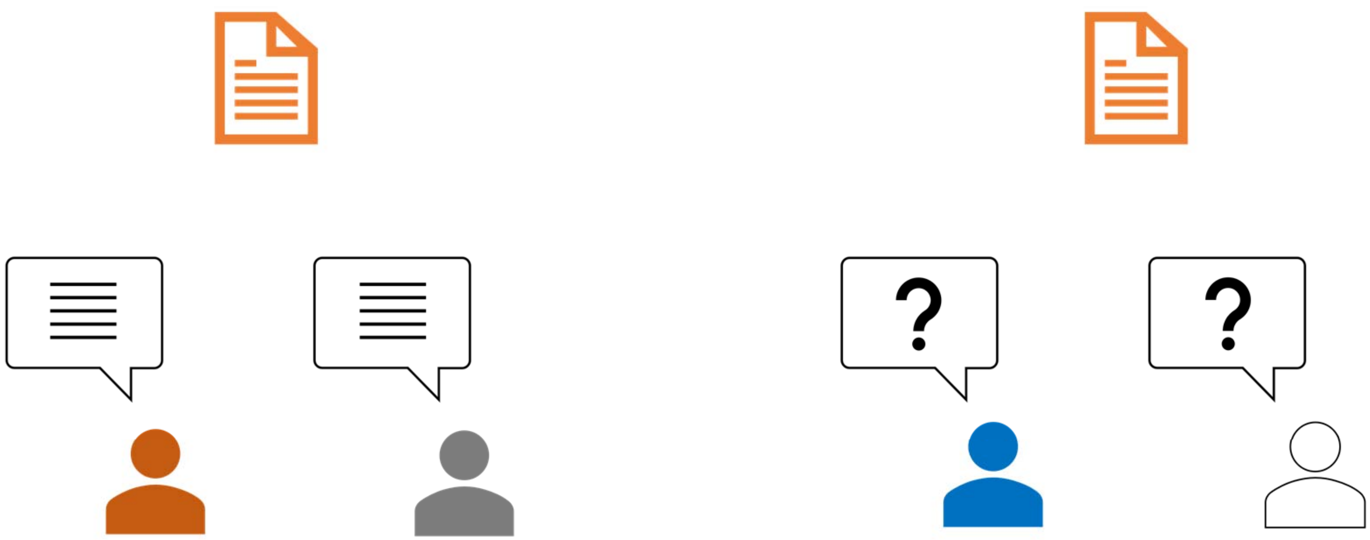


Key Idea 2: Item Exchange Graph
Given an allocation A, let $\mathcal{G}(A) = \langle O, E \rangle$ be a directed graph over the item set, where (o_1, o_2) exists if: $o_1 \in A_i$ and $v_i(A_i \setminus \{o_1\} \cup \{o_2\}) = v_i(A_i)$

Theorem [Schrijver, 2003; Barman and Verma, 2021; Viswanathan and Zick, 2022]: let $F_i(A) = \{o \in O \colon \Delta_i(A_i, o) = 1\}$, then there is a transfer path from i to A_0 iff there is a path from some $o \in F_i(A)$ to some $o' \in A_0$ in $\mathcal{G}(A)$.

Peer Review

Papers must be reviewed by suitable reviewers!



Reviewer Assignment Problem

Given *affinity scores*

	5	4	8	4
	9	6	1	8
	3	4	6	7

Reviewer Assignment Problem

Given *affinity scores*

5	4	8	4
9	6	1	8
3	4	6	7

- Widely used (OpenReview, CMT, etc)
- Roughly correlate with reviewer expertise and interest
- High affinity is good for papers & conferences

Reviewer Assignment Problem

Given *affinity scores*

5	4	8	4
9	6	1	8
3	4	6	7

Paper Requirements:

2

2

3

Reviewer Load Bounds:

≤ 2

≤ 3

≤ 1

≤ 2

Reviewers can be assigned 1x to any paper

Reviewer Assignment Problem

Given *affinity scores*

5	4	8	4
9	6	1	8
3	4	6	7

Paper Requirements:

2

2

3

Assume uniform for now, general case later

Reviewer Load Bounds:

≤ 2

≤ 3

≤ 1

≤ 2

Reviewers can be assigned 1x to any paper

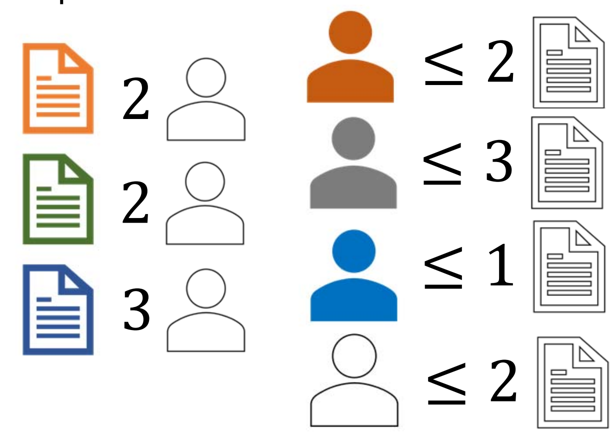
Reviewer Assignment Problem

Given *affinity scores*

5	4	8	4
9	6	1	8
3	4	6	7

Paper Requirements:

Reviewer Load Bounds:



Reviewers can be assigned 1x to any paper

Goal:
Allocation of reviewers to papers

Reviewer Assignment Problem

Given *affinity scores*

5	4	8	4
9	6	1	8
3	4	6	7

Goal:
Allocation of reviewers to papers

High affinity (welfare/USW)

Fast to compute

Adapts to new constraints

Reviewer Assignment Problem

Given *affinity scores*

5	4	8	4
9	6	1	8
3	4	6	7

Goal:
Allocation of reviewers to papers

High affinity (welfare/USW)

Fast to compute

Fair to papers!

Adapts to new constraints

Papers with poorly matched reviewers may get bad feedback, unfair rejection, etc

Reviewer Assignment Problem

Given *affinity scores*

5	4	8	4
9	6	1	8
3	4	6	7

Goal:
Allocation of reviewers to papers

High affinity (welfare/USW)

Fast to compute

Fair to papers!

Adapts to new constraints

Envy-freeness up to 1 Item (EF1)

- Large literature with simple/fast algorithms
- Any disparities in subjective value due to ≤ 1 reviewer

EF1 and Welfare

Theorem (Barman et al. 2019):
There is no polynomial-time m^{δ}-approximation for maximizing welfare over all EF1 allocations.

Justifies our use of a simple mechanism for EF1 allocations called *round-robin*

Proposition (following Aziz et al. 2015 and Aziz et al. 2016):
Maximizing welfare under round-robin is NP-hard

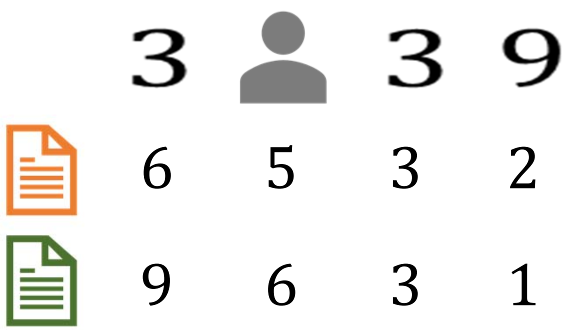
Goal:
Approximately maximize welfare under round-robin

Reviewer Round-Robin

Set an order over papers – papers are assigned 1 reviewer per round

Reviewer Round-Robin

Set an order over papers – papers are assigned 1 reviewer per round

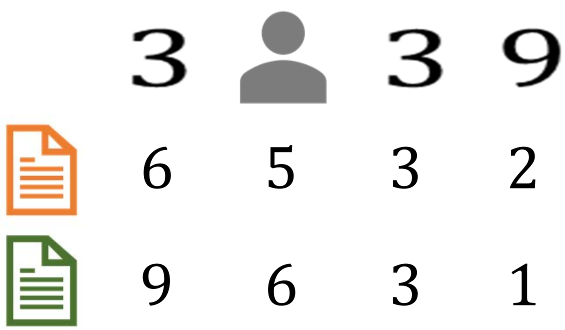


$USW = 16$

Reviewers assigned 1x to each paper ➔ Some matches disallowed to ensure EF1

Reviewer Round-Robin

Set an order over papers – papers are assigned 1 reviewer per round



$USW = 16$

$USW = 19$

Welfare depends on the order!

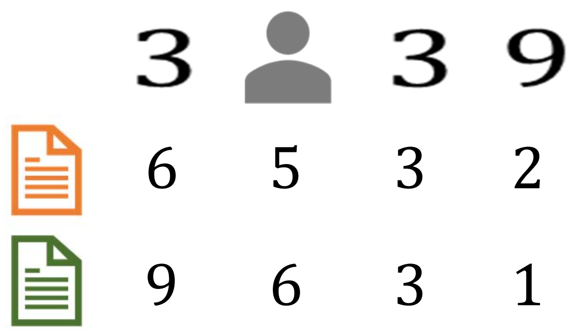
Greedy Reviewer Round-Robin

Put papers in order to maximize welfare for round-robin on a subset

			3	9
	6	5	3	2
	9	6	3	1

Greedy Reviewer Round-Robin

Put papers in order to maximize welfare for round-robin on a subset



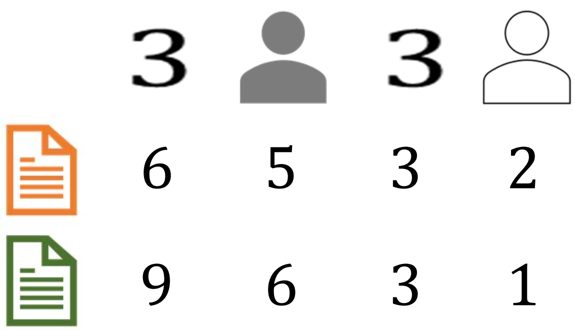
Greedy Reviewer Round-Robin

Put papers in order to maximize welfare for round-robin on a subset

	3		3	9
	6	5	3	2
	9	6	3	1

Greedy Reviewer Round-Robin

Put papers in order to maximize welfare for round-robin on a subset



$USW = 19$

Greedy Reviewer Round-Robin

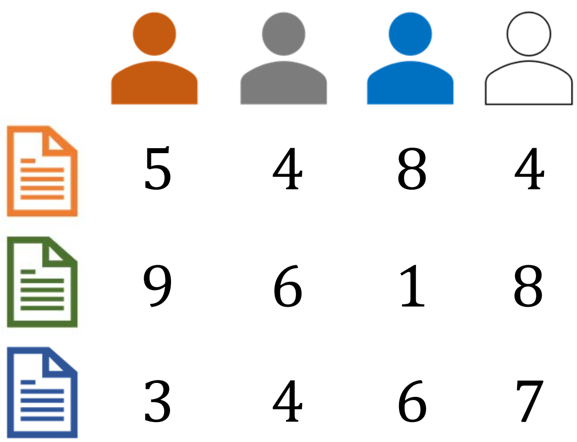
Theorem [Payan and Zick, IJCAI 2022]:
Greedy Reviewer Round-Robin gives a $1 + \gamma^2$ approximation to maximum welfare over orders, when the welfare as a function of the order is γ-weakly submodular.

Proof sketch:

- Consider sets of (paper, position) tuples
- Round-robin orders are equivalent to sets with exactly one tuple per paper and position
- Greedy Reviewer Round-Robin is the same as maximizing a monotonically increasing, γ-weakly submodular function over the intersection of two partition matroids

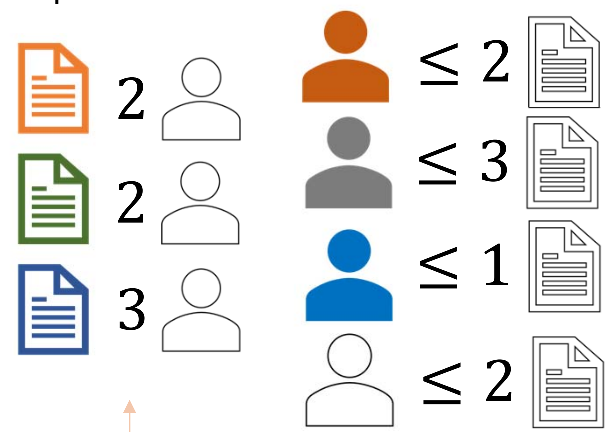
Non-Uniform Demands

Given *affinity scores*



Paper Requirements:

Reviewer Load Bounds:



Reviewers can be assigned 1x to any paper

Assumed uniform before, now general case

FairSequence

Assign to papers with lowest fraction of demand satisfied

Ties broken to greedily maximize affinity

Ensure *weighted* EF1

FairSequence

Assign to papers with lowest fraction of demand satisfied

Ties broken to greedily maximize affinity

Ensure *weighted* EF1

	8			5
0/2	5	4	8	4
0/2	9	6	1	8
0/3	3	4	6	7

FairSequence

Assign to papers with lowest fraction of demand satisfied

Ties broken to greedily maximize affinity

Ensure *weighted* EF1

		8			5
0/2		5	4	8	4
1/2		9	6	1	8
0/3		3	4	6	7

FairSequence

Assign to papers with lowest fraction of demand satisfied

Ties broken to greedily maximize affinity

Ensure *weighted* EF1

		8			5
0/2		5	4	8	4
1/2		9	6	1	8
0/3		3	4	6	7

FairSequence

Assign to papers with lowest fraction of demand satisfied

Ties broken to greedily maximize affinity

Ensure *weighted* EF1

		8			5
0/2		5	4	8	4
1/2		9	6	1	8
0/3		3	4	6	7

FairSequence

Assign to papers with lowest fraction of demand satisfied

Ties broken to greedily maximize affinity

Ensure *weighted* EF1

Etc…

No welfare guarantees, but good in practice
Much faster too!

Welfare and Fairness

Our Approaches

MIDL

	TPMS (OPT)	FairFlow	PR4A	GRRR	FairSeq
USW (% of OPT)	100%	100%	98%	98%	99%
# EF1 Viol.	0	0	0	0	0

CVPR

	TPMS (OPT)	FairFlow	PR4A	GRRR	FairSeq
USW (% of OPT)	100%	96%	94%	88%	92%
# EF1 Viol.	473545	23344	82	0	0

CVPR ’18

	TPMS (OPT)	FairFlow	PR4A	GRRR	FairSeq
USW (% of OPT)	100%	97%	97%	94%	96%
# EF1 Viol.	134	25	2	0	0

FairSequence Fits the Criteria

Fairness	GRRR and FairSeq are the only approaches that satisfy EF1
Welfare	High USW w.r.t. TPMS (OPT) and algorithms used in practice
Speed	>5x speedup compared to FairFlow/PR4A
Flexibility	Simplicity ➔ flexibility

Look for FairSequence!

Integrated into Open Review

Ask your conference organizer today if FairSequence is right for you!

References

- Babaioff et al. "Fair and Truthful Mechanisms for Dichotomous Valuations", AAAI 2021
- Benabbou et al. "Finding Fair and Efficient Allocations for Matroid Rank Valuations", ACM TEAC 2021
- Halpern et al. "Fair Division with Binary Valuations: One Rule to Rule Them All", WINE 2020
- Payan and Zick "I Will Have Order! Optimizing Orders for Fair Reviewer Assignment", IJCAI 2022