Case 2: Monotone chores. Does envy-cycle elimination work?

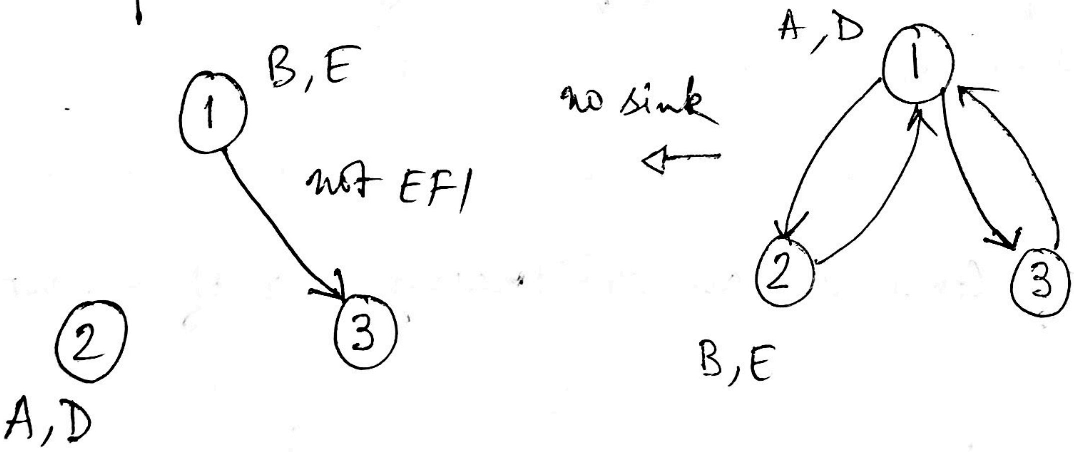
Envy-cycle elimination algorithm

While there is an unallocated chore
- if the envy graph has a sink vertex, assign the ~~go~~ chore to that agent
- ~~(x)~~ else, resolve any envy cycle until a sink shows up and assign the chore to it.

Ex:

	A	B	C	D	E
1	-1	-3	-2	-9	-3
2	0	-1	-4	-1	-2
3	0	-2	-1	0	-4

break ties
1 → 2 → 3
chore assignment order
A → B → C → D → E



Problem: EF1 for chores requires chores to be dropped from own bundle. When doing the envy cycle resolution, it gets a new bundle and its chores could be totally different.

In this example, original bundle had one large negative chores to ensure EF1. The exchanged bundle does not have any such chore.

Clearly arbitrary envy-cycle resolution won't work.

But there is an envy-cycle whose resolution would have worked. The other one.

The difference is that the other envy-cycle reduces envy by the maximum amount.

Max envy-cycle elimination algorithm (top-trading EC)

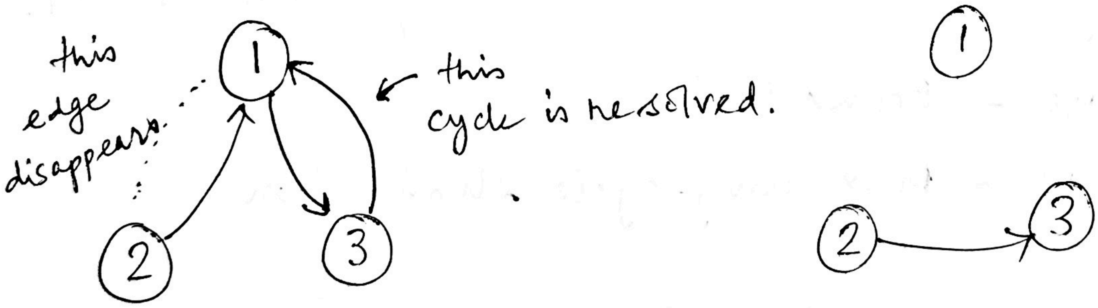
While there is an unallocated chore

- if the envy graph has a sink vertex, assign a chore to that agent
- else, resolve any max envy cycle until a sink appears and assign the chore to it.

The graph is now not allowing multiple outgoing edges. Allows only the edge that has max envy, i.e., each agent points to her favorite bundle.

If two bundles have same max value, point to both.

If there is no sink in this graph, a cycle must exist.

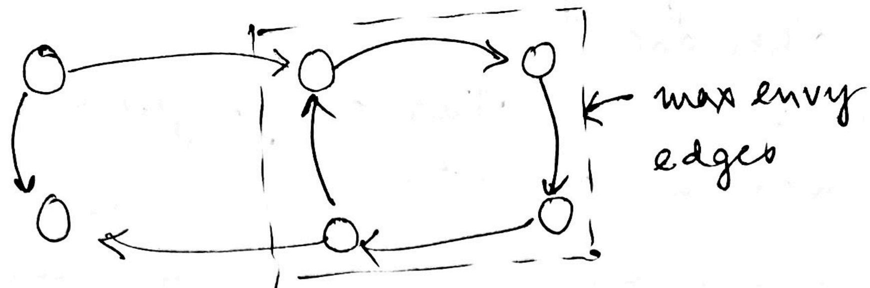


Why does max envy cycle elimination converge?

Similar argument as envy cycle elimination

- it converges in poly-time and make polynomial queries for the valuations.

Why is max envy cycle elimination EF1?



- Agents inside the cycle: their envy disappears to all other nodes - since they are getting their most preferred bundle - this has the highest value to them. Hence they won't envy others.
- Agents outside the cycle: their ~~incoming~~ outgoing envy edge can shift node, but the magnitude of their envy does not change. Also, they retain their bundles. Hence, the item they could drop earlier to ensure EF still is with them.

Thm: for monotone chores, the allocation computed by the top-trading envy cycle elimination algorithm is EF1.

Additive chores - Round Robin

Monotone chores - Max envy-cycle elimination

<u>Mixed items</u>: Goods for some, chore for others

e.g., noise level of rooms, ISMP/DAMP mentorship

	A	B	C
1	3	-1	$\boxed{-1}$
2	$\boxed{-4}$	$\boxed{1}$	-2

Envy-freeness notion:

- either drop a "good" from other agent's bundle, OR
- drop a "chore" from self bundle.

Envy-freeness upto one item (EF1)

An allocation $A = (A_1, A_2, \ldots, A_n)$ is Envy free upto one item if $\forall i, j$, $\exists x \in A_i \cup A_j$ s.t.

$$v_i(A_i \setminus \{x\}) \geq v_i(A_j \setminus \{x\}).$$

Observation: Simple Round Robin fails EF1 for additive mixed valuations.

	A	B
1	1	−1
2	1	−1

Variant of Round Robin: Double Round Robin Algorithm.

- Partition the items into positive and negative sets

  Positive: items with positive value for at least one agent (at least one agent consider it as a "good").

  Negative: all other items (considered a "chore" by all agents — all zero items are also in this category)

	A	B	C	D	E
1	−4	−1	−2	2	−4
2	0	−1	−5	−2	−1
3	−4	−2	−5	0	2
	negative			positive	

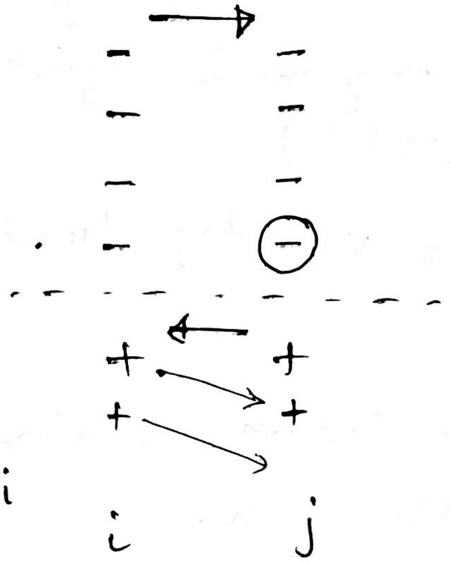
Doubly round robin algorithm

- fix an ordering $a_1, a_2, \dots, a_n$ over the agents
- run round robin allocation of the negative items in the order $a_1, a_2, \dots, a_n$.
- if the number of ~~items~~ negative items ~~are~~ is not a multiple of n, add dummy items (value zero for all agents) to make it a multiple of n. (those items get allocated first as a consequence unless someone has zero value for the real items)
- once all negative items are allocated, run round robin allocation of the positive items in the order $a_n, a_{n-1}, \dots, a_2, a_1$
- agents are allowed to skip turn if none of the remaining items are of positive value to them.

Question: Does it converge?

- Negative item allocation certainly does
- Positive item allocation can't have a skipping cycle
  - there exists at least one agent ~~to~~ that values an item positive, hence not every agent will skip turn.

Question: Does DRR return an EF1 allocation?



Phase 1 will be EF1 for both agents — from the EF1 argument for chores. These are chores for every agent.
i never envies j
j may envy upto the last chore.

In phase 2:
j picks first
has no envy to i
i can envy j
but upto the first "good" j gets.

Another class of valuations

Doubly monotone:

Each agent can partition each item as a "good", or a "chore" — not necessarily in the additive form rather in the marginal form

- $v_i(S \cup \{g\}) - v_i(S) \geq 0 \quad \forall S$, then g is a good for i
- $v_i(S \cup \{c\}) - v_i(S) \leq 0 \quad \forall S$, then c is a chore for i.

Envy-cycle + Top trading Algorithm

(1) Partition items into positive and negative groups
- positive: "good" by at least one agent
- negative: "chore" by all agents.

11-7

(2) Assign positive items via envy-cycle elimination (envy graph defined only on the agents who consider it as a "good")

(3) Assign negative items via top trading envy-cycle elimination

Thm: For doubly monotone items, the above algorithm returns an EF1 allocation.

EF1

Goods
- additive : Round Robin
- monotone nondecreasing : envy cycle elimination
  add PO → Nash optimal

Bads
- additive : Round Robin
- monotone nonincreasing : max envy-cycle elimination
  add PO : known for few special type of valuation, e.g., bivalued.

Mixed
- additive : Doubly round robin
- doubly monotone : Envy cycle + max envy cycle
- general : open