Lecture 3: Fair Division of Indivisibles (Part 1)

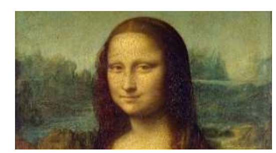
CS 580

Instructor: Ruta Mehta

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Fair Division

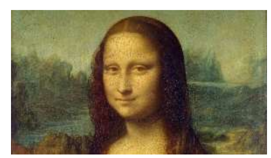
Scares resources



Goal: allocate *fairly and efficiently.*

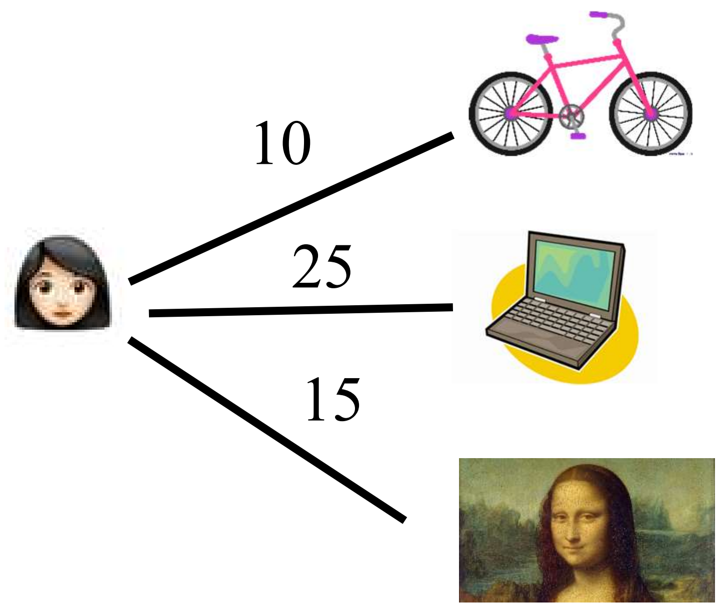
And do it quickly (fast algorithm)!

- n agents: 1 ,…, n,
- M: set of m indivisible items (like cell phone, painting, etc.)



- Agent i has a valuation function $v_i : 2^m \rightarrow \mathbb{R}$ over subsets of items
  - Monotone: the more the happier

Additive Valuations: $v_i(S) = \sum_{j \in S} v_{ij}$



- n agents: 1 ,…, n,
- M: set of m indivisible items (like cell phone, painting, etc.)
- Agent i has a valuation function $v_i : 2^m \rightarrow \mathbb{R}$ over subsets of items
  - Monotone: the more the happier
- Goal: Find a *fair* allocation

Fairness:

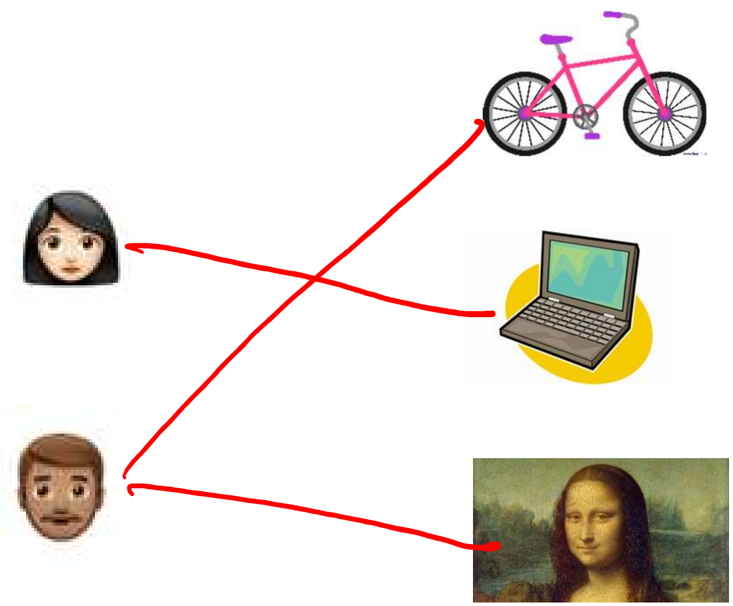
Envy-free (EF): no one *envies* other's bundle

Proportional (Prop): each agent i gets at least $\frac{v_i(M)}{n}$ $\rightarrow v_i\left(\frac{M}{n}\right)$

Allocations, and their value

[10, 25, 15]

[10, 20, 10]



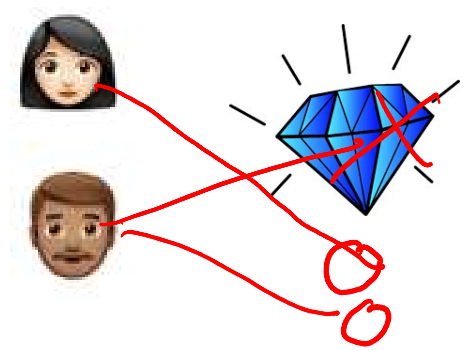
- n agents: 1 ,…, n,
- M: set of m indivisible items (like cell phone, painting, etc.)
- Agent i has a valuation function $v_i : 2^m \rightarrow \mathbb{R}$ over subsets of items
  - Monotone: the more the happier
- Goal: Find a *fair* allocation

Fairness:

Envy-free (EF): no one *envies* other's bundle

Proportional (Prop): each agent i gets at least $\frac{v_i(M)}{n}$

Neither exists!



Plan

- EF1: EF up to one item
  - Round-Robin algorithm
  - Envy-cycle elimination algorithm
- Stronger notions + Open questions
  - “Good” EF1 allocations: EF1 + Pareto optimal
  - EFX: EF up to *any* item
- Prop1: Prop up to one item
  - Algorithm through CE. PO in addition.

Envy-Freeness for Indivisibles

EF up to One Item (EF1) [B11]

- An allocation $(A_1, \dots, A_n)$ is EF1 if for every agent i

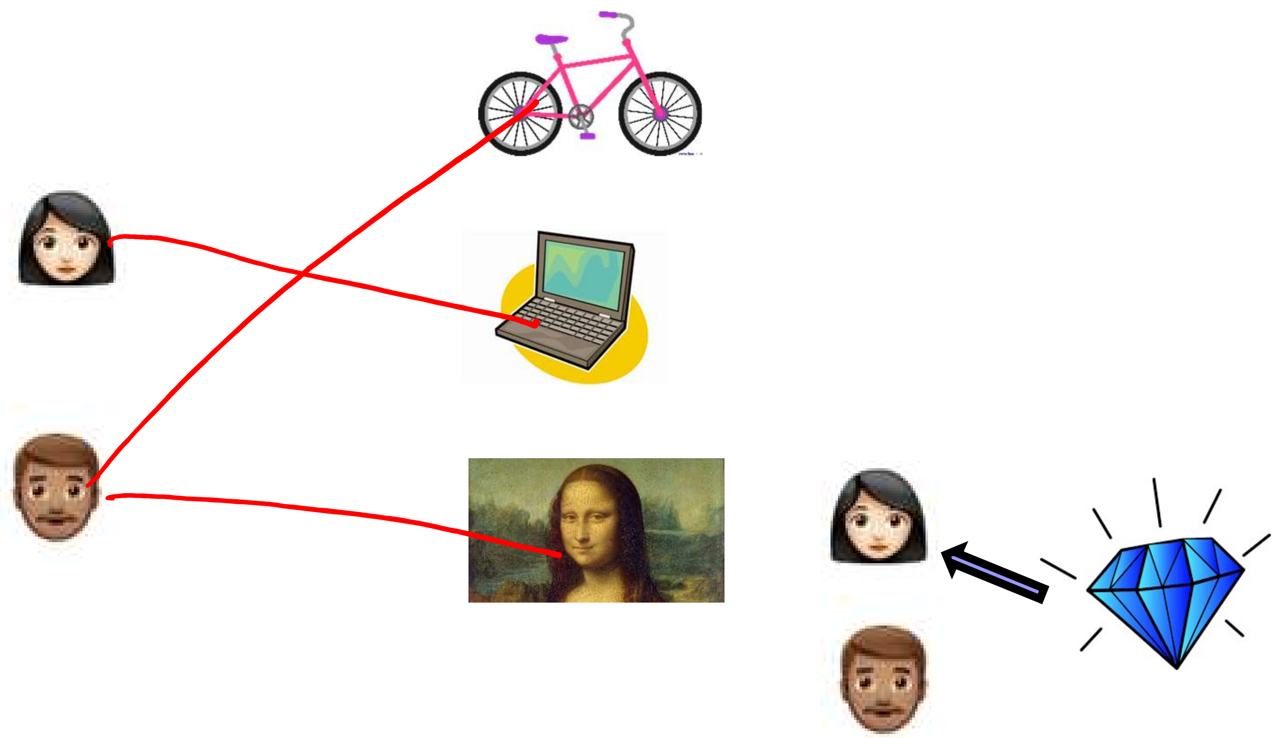
$$\forall k \in N, \qquad v_i(A_i) \geq v_i(A_k \setminus g), \qquad \exists g \in A_k$$

That is, agent i may envy agent k, but the envy can be eliminated if we remove a single item from $k's$ bundle

Envy-Freeness up to One Item (EF1) [B11]

[15, 25, 15]

[11, 20, 10]



Fast Algorithms for EF1

Round Robin Algorithm (Additive)

- Fix an ordering of agents arbitrarily
- While there is an item unallocated
  - i: next agent in the round robin order
  - Allocate i her most valuable item among the unallocated ones

	g_1	g_2	g_3	g_4	g_5
a_1	10	15	9	8	3
a_2	10	8	15	9	4
a_3	10	9	8	15	5

	$R1$	$R2$
a_1		
a_2		
a_3		

Theorem. The final allocation is EF1.

Round Robin Algorithm (Additive)

- Fix an ordering of agents arbitrarily
- While there is an item unallocated
  - i: next agent in the round robin order
  - Allocate i her most valuable item among the unallocated ones

Observation 1: First agent does not envy anyone!

Round Robin Algorithm (Additive)

- Fix an ordering of agents arbitrarily
- While there is an item unallocated
  - i: next agent in the round robin order
  - Allocate i her most valuable item among the unallocated ones

Observation 2: For the ith agent, if we remove first $(i-1)$ items allocated to first $(i-1)$ agents respectively, then the allocation is envy-free for agent i.

Round Robin Algorithm (Additive)

- Fix an ordering of agents arbitrarily
- While there is an item unallocated
  - i: next agent in the round robin order
  - Allocate i her most valuable item among the unallocated ones

Observation 1: First agent does not envy anyone!

Observation 2: For the ith agent, if we remove first $(i-1)$ items allocated to first $(i-1)$ agents respectively, then the allocation is envy-free for agent i.

Theorem. Round Robin Algorithm gives an EF1 allocation when v_is are additive.

General Monotone Valuations: Envy-Cycle Procedure [LMMS04]

- General Monotonic Valuations: $v_i(S) \leq v_i(T),\ \forall S \subseteq T \subseteq M$

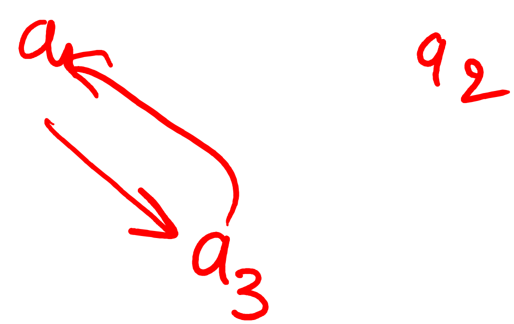
  (M: Set of all items)

Envy-Cycle Procedure (General) [LMMS04]

- General Monotonic Valuations: $v_i(S) \leq v_i(T),\ \forall S \subseteq T \subseteq M$
- **Partial allocation:** $(A_1, \ldots, A_n)$ where $\cup_i A_i \subseteq M$

- **Envy-graph** of a partial allocation $(A_1, \ldots, A_n)$
  - Vertices = Agents
  - Directed edge (i, i') if i **envies** i' (i.e., $v_i(A_i) < v_i(A_{i'})$)

	g_1	g_2	g_3	g_4	g_5
a_1	10	15	9	8	3
a_2	10	8	15	9	4
a_3	10	9	8	15	5



Envy-Cycle Procedure (General) [LMMS04]

- General Monotonic Valuations: $v_i(S) \leq v_i(T),\ \forall S \subseteq T \subseteq M$
- Envy-graph of a partial allocation $(A_1, \ldots, A_n)$ where $\cup_i A_i \subseteq M$
  - Vertices = Agents
  - Directed edge (i, i') if i **envies** i' (i.e., $v_i(A_i) < v_i(A_{i'})$)

- **Main Observation:**

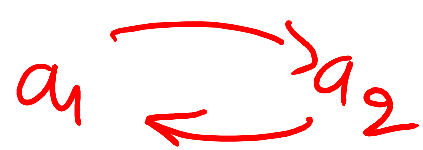
  Agent i is a *source* in the envy-graph ⇒ No one envies agent i

- **Idea!** Allocate one item at a time, maintaining EF1 property.
  - Given a partial EF1 allocation, construct its envy-graph and assign one unallocated item, say j, to a source agent, say i, and the resulting allocation is still EF1!
  - No agent envies i if we remove item j from her bundle

If there is no source in envy-graph, then?

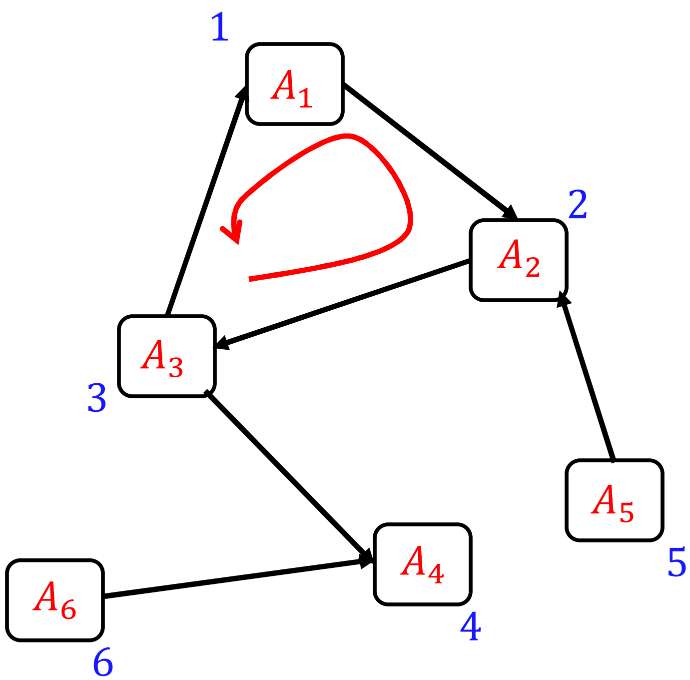
- there must be cycles
- How to eliminate them?

	g_1	g_2	g_3	g_4	g_5
a_1	10	15	9	8	3
a_2	10	8	15	9	4
a_3	10	9	8	15	5

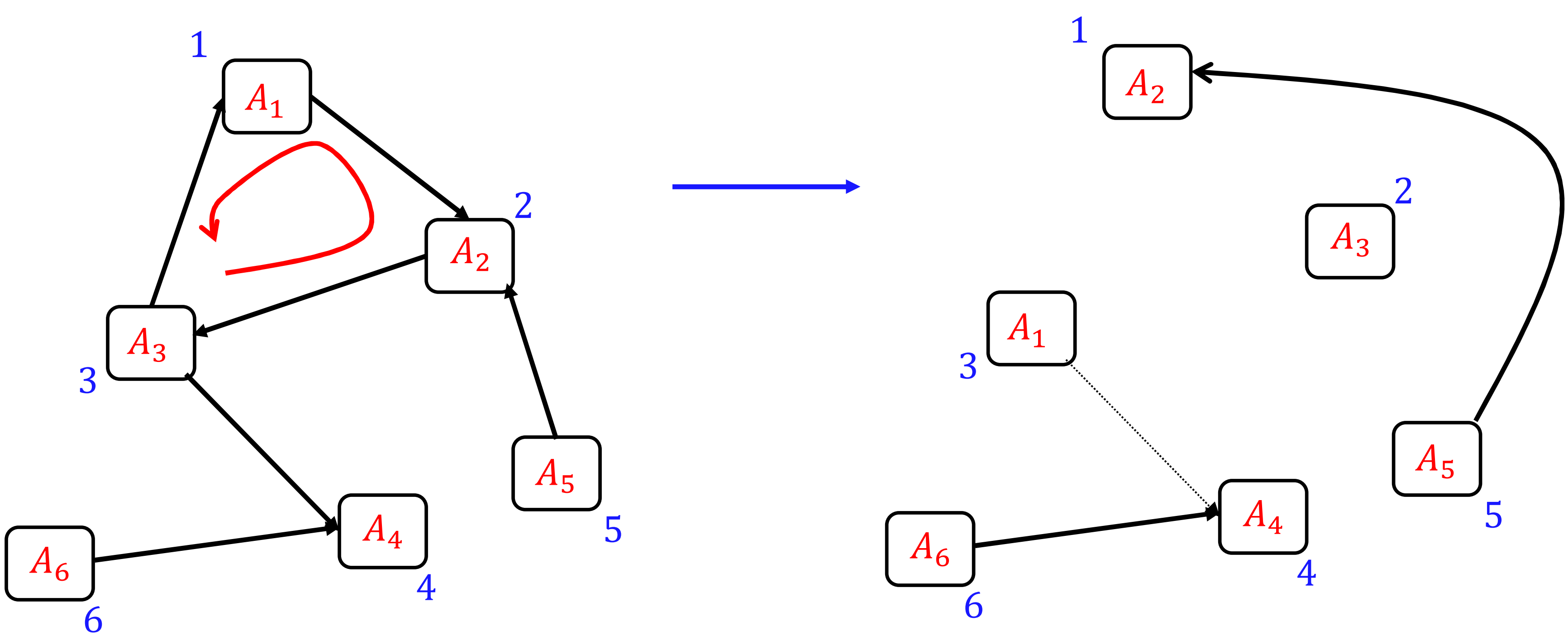


If there is no source in envy-graph, then?

- there must be cycles
- How to eliminate them?



- If there is no source in envy-graph, then
  - there must be cycles
  - How to eliminate them?
- **Cycle elimination:** rotate bundles along the cycle.



- If there is no source in envy-graph, then
  - there must be cycles

Cycle elimination: rotate bundles along the cycle.

- EF1?
  - Can valuation of any agent decrease?

- If there is no source in envy-graph, then
  - there must be cycles

Cycle elimination: rotate bundles along the cycle.

- EF1?
  - Can valuation of any agent decrease?

  NO! Agents on an eliminated cycle gets better off, others remain same.

  - Can there be new envy edges?

  NO! The bundles remain the same – We are only changing their owners!

  Hence, no new envies are formed.

Claim 1. After every cycle elimination, the allocation remains EF1.

- If there is no source in envy-graph, then
  - there must be cycles

Cycle elimination: rotate bundles along the cycle.

Claim 1. After every cycle elimination, the allocation remains EF1.

Keep eliminating cycles by exchanging bundles along a cycle until there is a source.

- Termination?
  - Number of edges decrease after each cycle elimination.

Claim 2. The process terminates in at most O(#edges) many cycle eliminations.

Envy-Cycle Procedure [LMMS04]

$A \leftarrow (\emptyset, \ldots, \emptyset)$

$R \leftarrow M$ // unallocated items

While $R \neq \emptyset$

- If envy-graph has no source, then there must be cycles
- Keep removing cycles by exchanging bundles along a cycle, until there is a source
- Pick a source, say i, and allocate one item g from R to i

  $(A_i \leftarrow A_i \cup g;\ R \leftarrow R \setminus g)$

Output A

- Running Time? 

Proportional (average)

- n agents
- M: set of m indivisible items (like cell phone, painting, etc.)
- Agent i has a valuation function $v_i : 2^m \rightarrow \mathbb{R}$ over subsets of items

Fairness:

Envy-free (EF)

Proportional (Prop):

Get value at least average of the grand-bundle

$$v_i(A_i) \geq \frac{1}{n} v_i(M)$$

	g_1	g_2	g_3	g_4
a_1	100	100	10	90
a_2	100	100	90	10

Sub-additive Valuations

Sub-additive:

$$v_i(A \cup B) \le v_i(A) + v_i(B), \qquad \forall A, B \in M$$

Claim: $EF \Rightarrow Prop$

Proof:

Prop: May not always exist!

- n agents
- M: set of m indivisible items (like cell phone, painting, etc.)
- Agent i has a valuation function $v_i : 2^m \to \mathbb{R}$ over subsets of items

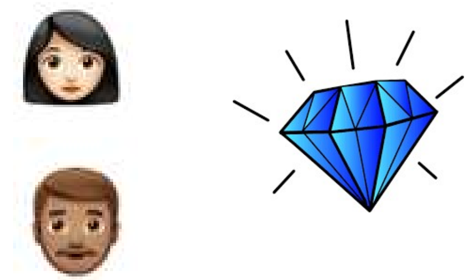
Fairness:

Envy-free (EF)

Proportional (Prop):

Get value at least average of the grand-bundle

$$v_i(A_i) \geq \frac{1}{n} v_i(M)$$

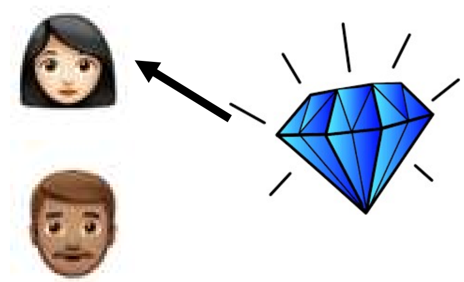


Proportionality up to One Item (Prop1)

- Prop1: A is proportional up to one item if each agent gets at least $1/n$ share of all items after adding one more item from outside:

$\forall i \in N$,

$$v_i(A_i \cup \{g\}) \geq \frac{1}{n} v_i(M), \qquad \exists g \in M \setminus A_i,$$ ~~$\forall i \in N$~~



Prop1

Claim: EF1 implies Prop1 for additive valuations

Proof: EF1: $\forall i, \forall k,\quad v_i(A_i) \geq v_i(A_k \setminus g) \qquad \exists g \in A_k$

$$= v_i(A_k) - v_i(g) \quad (\because v_i \text{ additive})$$

$$\geq v_i(A_k) - \max_{g \in M \setminus A_i} v_i(g)$$

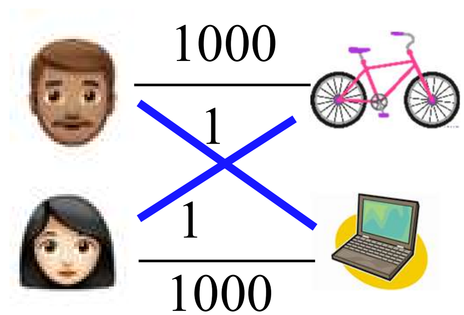
Hence $\forall i,\ n\, v_i(A_i) \geq \sum_{k=1}^{n} v_i(A_k) - n \max_{g \in M \setminus A_i} v_i(g)$

$$\Rightarrow v_i(A_i) + \max_{g \in M \setminus A_i} v_i(g) \geq \frac{v_i(M)}{n} \quad (\because v_i \text{ additive})$$

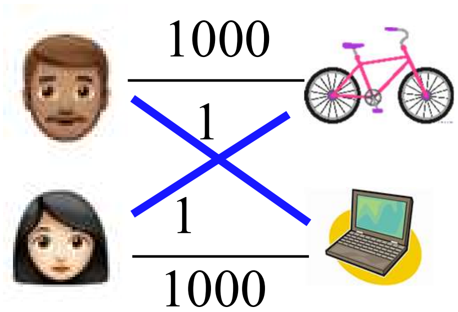
$$\Rightarrow v_i(A_i \cup \{g\}) \geq \frac{v_i(M)}{n}, \quad \exists g \in M \setminus A_i$$

∎

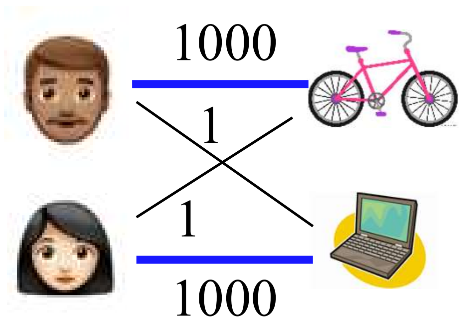
How Good is an EF1 or Prop1 Allocation?



How Good is an EF1 or Prop1 Allocation?



- Certainly not desirable!



“Good” EF1/Prop1 Allocation: Pareto Optimality

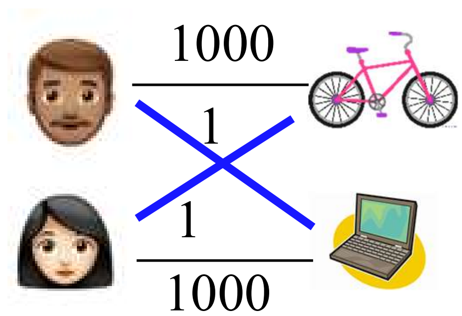
- Issue: Many EF1/Prop1 allocations!
- We want an algorithm that outputs a good EF1/Prop1 allocation

Pareto optimal (PO): No other allocation is better for all

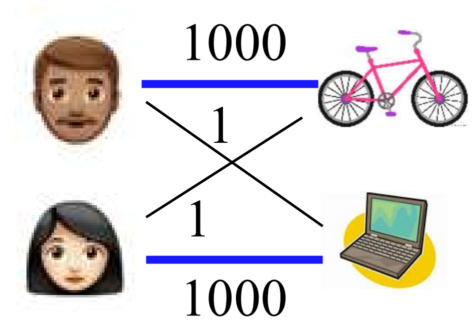
- An allocation $Y = (y_1, y_2, \dots, y_n)$ Pareto dominates another allocation $X = (x_1, x_2, \dots, x_n)$ if
  - $v_i(y_i) \geq v_i(x_i)$, for all buyers i and
  - $v_k(y_k) > v_k(x_k)$ for some buyer k

- X is said to be Pareto optimal (PO) if there is no Y that Pareto dominates it

How Good is an EF1 or Prop1 Allocation?



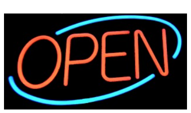
PO



“Good” EF1 Allocation: EF1+PO

- Issue: Many EF1 allocations!
- We want an algorithm that outputs a good EF1 allocation
  - Pareto optimal (PO)

- Goal: EF1 + PO allocation
- Existence?
  - NO [CKMPS14] for general (subadditive) valuations
  - YES for additive valuations [CKMPS14]



submodular valuations

“Good” EF1 Allocation: EF1+PO

- Issue: Many EF1 allocations!
- We want an algorithm that outputs a good EF1 allocation
  - Pareto optimal (PO)

- Goal: EF1 + PO allocation
- Existence?
  - NO [CKMPS14] for general (subadditive) valuations
  - YES for additive valuations [CKMPS14] Computation?

submodular valuations

EF1+PO (Additive)

- Computation: pseudo-polynomial time algorithm [BKV18]

  Complexity of finding an EF1+PO allocation

- Difficulty: Deciding if an allocation is PO is co-NP-hard [KBKZ09]

EF1+PO (Additive)

- Computation: pseudo-polynomial time algorithm [BKV18]

  Complexity of finding an EF1+PO allocation

- Difficulty: Deciding if an allocation is PO is co-NP-hard [KBKZ09]

- Approach: Achieve EF1 while maintaining PO
  - PO certificate: competitive equilibrium!

Prop1 + PO

- EF1 implies Prop1 for additive valuations

  $\Longrightarrow$ Round Robin outputs a Prop1 allocation. But need not be PO!

- Prop1+PO: Additive Valuations
  - EF1 + PO allocation exists ⇒ Prop1 + PO exists.
    - but no polynomial-time algorithm is known!
  - Prop1 + PO Computation?
    - Algorithm based on competitive equilibrium (HW).

EFX: Envy-free up to *any* item

Envy-Freeness up to One Item (EF1)

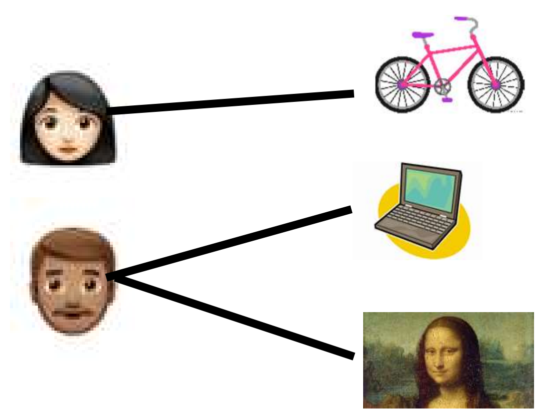
- An allocation $(A_1, \ldots, A_n)$ is EF1 if for every agent i

$$\forall k \in N, \qquad v_i(A_i) \geq v_i(A_k \setminus g), \qquad \exists g \in A_k.$$

That is, agent i may envy agent k, but the envy can be eliminated if we remove a single item from $k's$ bundle

[10, 10, 20]

[1, 20, 10]



Envy-Freeness up to Any Item (EFX) [CKMPS14]

- An allocation $(A_1, \dots, A_n)$ is EFX if for every agent i

$$\forall k \in N, \qquad v_i(A_i) \geq v_i(A_k \setminus g), \qquad \forall g \in A_k.$$

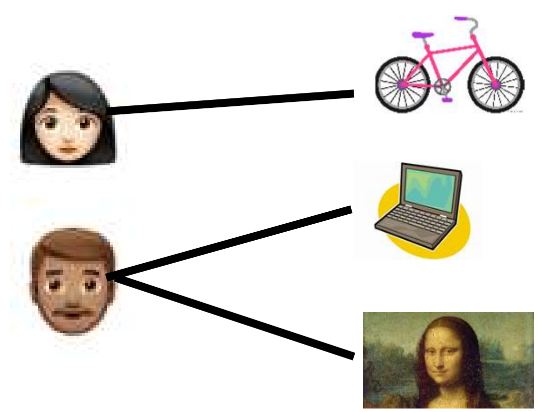
That is, agent i may envy agent k, but the envy can be eliminated if we remove any single item from $k's$ bundle

EF1

EFX ?

[10, 10, 20]

[1, 20, 10]



EFX: Existence

- General Valuations [PR18]
  - $n = 2$
  - Identical Agents

- Additive Valuations
  - $n = 3$ [CGM20]

Additive ($n > 3$), General ($n > 2$)

"Fair division's biggest problem" [P20]

Summary

Covered

- EF1 (existence/polynomial-time algorithm)
- EF1 + PO (partially)
- EFX (partially)
- Prop1

Not Covered

- EFX for 3 (additive) agents
- Partial EFX allocations
  - Little Charity [CKMS20, CGMMM21]
  - High Nash welfare [CGH19]
- Chores
  - EF1 (existence/ polynomial-time algorithm)

Major Open Questions (additive valuations)

- EF1+PO: Polynomial-time algorithm
- EF1+PO: Existence for chores
- EFX : Existence / Non-existence

References (Indivisible Case).

[BKV18] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In: *EC 2018*

[B11] Eric Budish. "The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes". In: *J. Political Economy* 119.6 (2011)

[CKMPSW14] Ioannis Caragiannis, David Kurokawa, Herve Moulin, Ariel Procaccia, Nisarg Shah, and Junxing Wang. "The Unreasonable Fairness of Maximum Nash Welfare". In: *EC* 2016

[CGH20] Ioannis Caragiannis, Nick Gravin, and Xin Huang. Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In: *EC 2019*

[CGM20] Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn: EFX Exists for Three Agents**.** In: *EC 2020*

[CGMMM21] Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, Ruta Mehta, Pranabendu Misra: Improving EFX Guarantees through Rainbow Cycle Number. In: EC 2021.

[CKMS20] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. In: *SODA 2020*

[KBKZ09] Bart de Keijzer, Sylvain Bouveret, Tomas Klos, and Yingqian Zhang. "On the Complexity of Efficiency and Envy-Freeness in Fair Division of Indivisible Goods with Additive Preferences". In: *Algorithmic Decision Theory (ADT)*. 2009

[LMMS04] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. "On approximately fair allocations of indivisible goods". In: *EC 2004*

[PR18] Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. In: *SODA 2018*

[P20] Ariel Procaccia: An answer to fair division's most enigmatic question: technical perspective. In: *Commun. ACM 63(4): 118 (2020)*