Fair Division of Indivisible Items

Jugal Garg

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

21st Max Planck Advanced Course on the Foundations of Computer Science
(ADFOCS)
August 24-28, 2020

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

- An allocation $(A_1, \dots, A_n)$ is EF1 if

$$v_i(A_i) \geq v_i\left(A_j \setminus g\right), \ \ \exists g \in A_j, \ \ \forall i, j$$

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

Scaling Valuations with Prices

- Envy-freeness is scale-free
- (A, p): CE
- Let's scale $v_{ij} \leftarrow v_{ij} \cdot \min_k \frac{p_k}{v_{ik}}$

$\Longrightarrow v_{ij} \leq p_j$ and $v_{ij} = p_j$ if $j \in A_i$

Prices can be treated as valuations at CE!

Price-Envy-Free (additive) [BKV18]

- (A, p): CE
- A is EF1 if $\quad v_i(A_i) \geq v_i(A_j \setminus g), \quad g \in A_j, \quad \forall i, j$

$$v_i(A_i) = p(A_i) \qquad p(A_j \setminus g) \geq v_i(A_j \setminus g), \quad \exists g \in A_j, \;\; \forall i, j$$

- A is Price-EF1 (pEF1) if

$$p(A_i) \geq p(A_j \setminus g), \qquad \exists g \in A_j, \qquad \forall i, j$$

- pEF1 ⇒ EF1 + PO

- (A, p): CE
- A is EF1 if $\quad v_i(A_i) \geq v_i(A_j \setminus g), \quad g \in A_j, \quad \forall i, j$

$$v_i(A_i) = p(A_i) \quad p(A_j \setminus g) \geq v_i(A_j \setminus g), \quad \exists g \in A_j, \quad \forall i, j$$

- A is Price-EF1 (pEF1) if

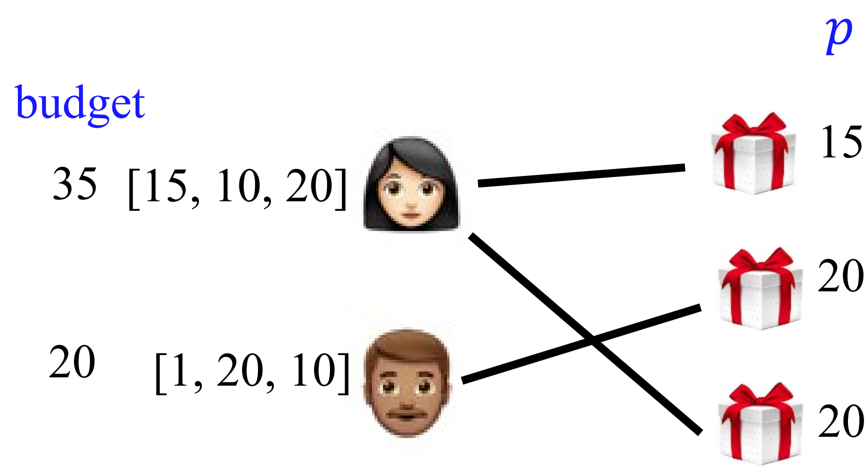
$$p(A_i) \geq p(A_j \setminus g), \quad \exists g \in A_j, \quad \forall i, j$$

- pEF1 ⇒ EF1 + PO

pEF1?

$$35 = p(A_1) > p(A_2 \setminus g_2) = 0$$

$$20 = p(A_2) > p(A_1 \setminus g_3) = 15$$



Theorem [BKV18]: There exists a pseudo-polynomial time procedure to find a pEF1 allocation

- (A, p): CE
- A is pEF1 if

$$p(A_i) \geq p(A_j \setminus g), \qquad \exists g \in A_j, \qquad \forall i, j$$

- If $\min_i p(A_i) \geq \max_j \min_{g \in A_j} p(A_j \setminus g)$ then ?

Theorem [BKV18]: There exists a pseudo-polynomial time procedure to find a pEF1 allocation

- (A, p): CE
- A is pEF1 if

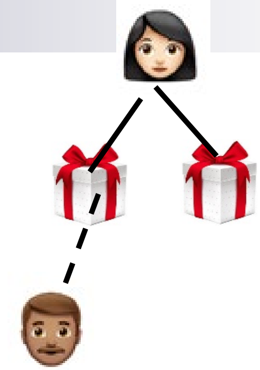
$$p(A_i) \geq p(A_j \setminus g), \qquad \exists g \in A_j, \qquad \forall i, j$$

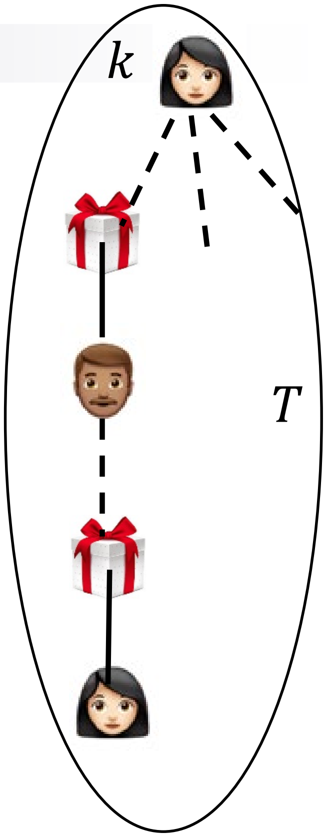
- If $\min_i p(A_i) \geq \max_j \min_{g \in A_j} p(A_j \setminus g)$ then ?

$\min_i p(A_i)$
(least spender)

$\max_j \min_{g \in A_j} p(A_j \setminus g)$
(big spender)

Procedure [BKV18]

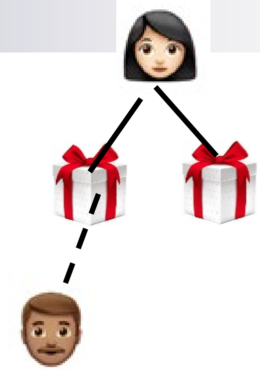




While A is not pEF1

$k \leftarrow \arg\min_i p(A_i)$ //least spender

$T \leftarrow$ Agents and items, k can reach in MBB residual network



While A is not pEF1

$k \leftarrow \arg\min_i p(A_i)$ //least spender

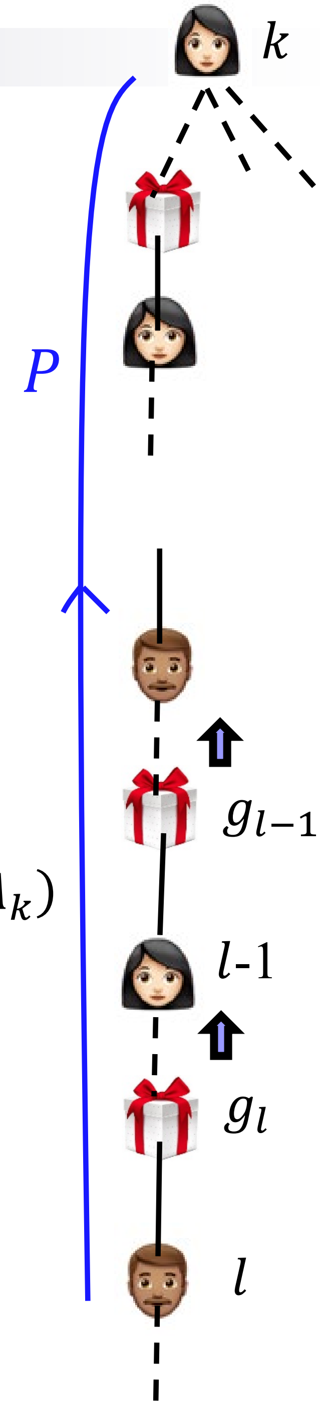
$T \leftarrow$ Agents and items, k can reach in MBB residual network

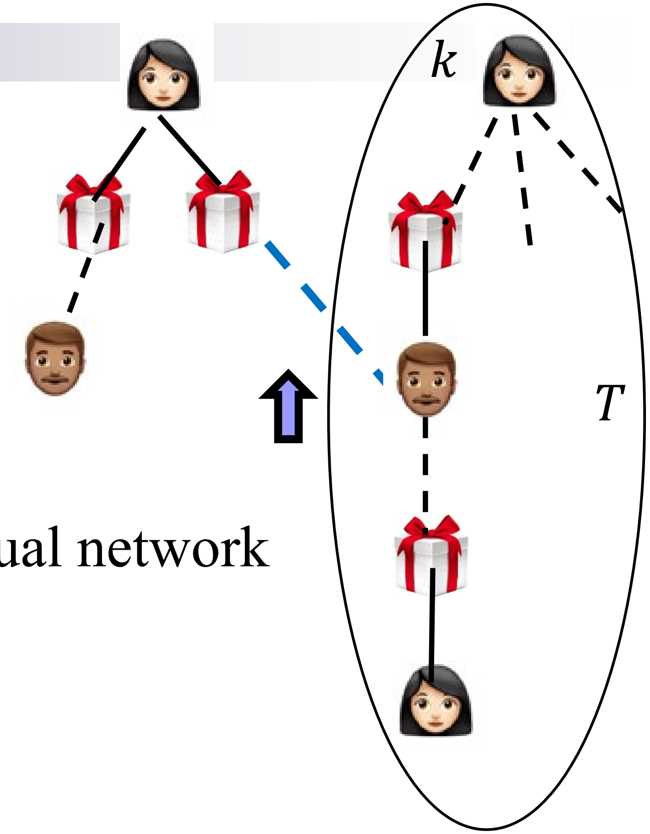
If k can reach l in T such that $p(A_l \setminus g_l) > p(A_k)$

Pick the nearest such l

$P \leftarrow$ Path from l to k

$A \leftarrow$ Reassign items along P until $p\big((A_j \cup g_{j+1}) \setminus g_j\big) \leq p(A_k)$





While A is not pEF1

$k \leftarrow \arg\min_i p(A_i)$ //least spender

$T \leftarrow$ Agents and items, k can reach in MBB residual network

If k can reach l in T such that $p(A_l \setminus g_l) > p(A_k)$

Pick the nearest such l

$P \leftarrow$ Path from l to k

$A \leftarrow$ Reassign items along P until $p\big((A_j \cup g_{j+1}) \setminus g_j\big) \leq p(A_k)$

else increase prices of items in T by a same factor until

Event 1: new MBB edge

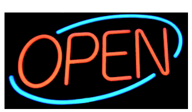
Event 2: k is not least spender anymore

Event 3: A becomes pEF1

$$\min_i p(A_i) \qquad \qquad \max_j \min_{g \in A_j} p(A_j \setminus g)$$

Lemma: The procedure converges to a pEF1 allocation in finite time!

Pseudo-polynomial time: Round $v_{ij}s$ to the nearest integer powers of $(1+\epsilon)$ for a suitably small $\epsilon > 0$ and then run the procedure



Complexity of finding an EF1+PO allocation!

New Fairness Notions

- n agents, m indivisible items (like cell phone, painting, etc.)
- Each agent i has a valuation function over subset of items denoted by $v_i : 2^m \rightarrow \mathbb{R}$
- Goal: fair and efficient allocation

Fairness:
  Envy-free (EF)
  Proportionality (Prop)

Efficiency:
  Pareto optimal (PO)

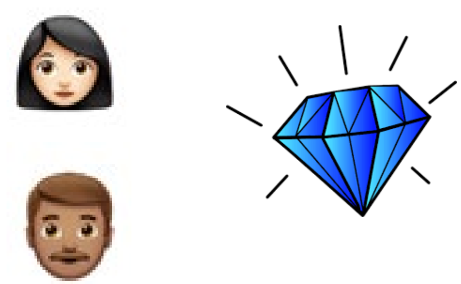
Maximum Nash Welfare (MNW)

EF1 EFX	Lecture 3
Prop1 MMS	Lecture 4
Guarantees	Lecture 5

Proportionality up to One Item (Prop1)

- A set N of n agents, a set M of m indivisible items
- Proportionality (Prop): Allocation $A = (A_1, \dots, A_n)$ is proportional if each agent gets at least $1/n$ share of all items:

$$v_i(A_i) \geq \frac{v_i(M)}{n}, \qquad \forall i \in N$$



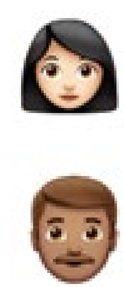
Proportionality up to One Item (Prop1)

- A set N of n agents, a set M of m indivisible items
- Prop: $A = (A_1, \dots, A_n)$ is proportional if each agent gets at least $1/n$ share of all items:

$$v_i(A_i) \geq \frac{1}{n} v_i(M), \qquad \forall i \in N$$

- 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:

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



Prop1

- EF1 implies Prop1 for subadditive valuations EXERCISE

  ⟹ Envy-cycle procedure outputs a Prop1 allocation

- Additive Valuations
  - EF1 + PO allocation exists but no polynomial-time algorithm is known!
  - Prop1 + PO?

Prop1 + PO [BK19]

- (p, x): CEEI
- x is envy-free ⇒ proportional
- we can assume that support of x is a forest (set of trees)
- In each tree:
  - Make some agent the root
  - Assign each item to its parent agent

Claim: The output of the above algorithm is Prop1 + PO

Prop1 + PO [BK19]

- (p, x): CEEI
- x is envy-free ⇒ proportional
- we can assume that support of x is a forest (set of trees)
- In each tree:
  - Make some agent the root
  - Assign each item to its parent agent

Claim: The output of the above algorithm is Prop1 + PO

Fairness Notions for Indivisible Items

- n agents, m indivisible items (like cell phone, painting, etc.)
- Each agent i has a valuation function over subset of items denoted by $v_i : 2^m \rightarrow \mathbb{R}$
- Goal: fair and efficient allocation

Fairness:

Envy-free (EF)

Proportionality (Prop)

Efficiency:

Pareto optimal (PO)

Maximum Nash Welfare (MNW)

EF1 EFX	Lecture 3
Prop1 MMS	Lecture 4
Guarantees	Lecture 5

Proportionality

- A set N of n agents, a set M of m indivisible items
- Proportionality: Allocation $A = (A_1, \dots, A_n)$ is proportional if each agent gets at least $1/n$ share of all items:

$$v_i(A_i) \geq \frac{v_i(M)}{n}, \qquad \forall i \in N$$

Maximin Share (MMS) [B11]

- Suppose we allow agent i to propose a partition of items into n bundles with the condition that i will choose at the end
- Clearly, i partitions items in a way that maximizes the value of her least preferred bundle
- μ_i := Maximum value of $i's$ least preferred bundle

Maximin Share (MMS) [B11]

- Suppose we allow agent i to propose a partition of items into n bundles with the condition that i will choose at the end
- Clearly, i partitions items in a way that maximizes the value of her least preferred bundle
- μ_i := Maximum value of $i's$ least preferred bundle

- Π := Set of all partitions of items into n bundles
- $\mu_i \coloneqq \max_{A \in \Pi} \min_{A_k \in A} v_i(A_k)$

- MMS Allocation: A is called MMS if $v_i(A_i) \geq \mu_i, \forall i$

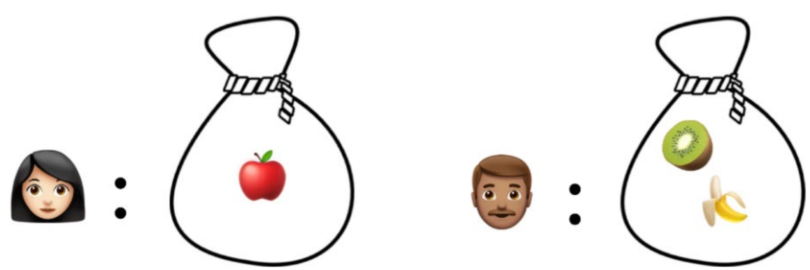
MMS value/partition/allocation

Assume additive valuations

Agent\Items	🍎	🍌	🥝
👩	3	1	2
👨	4	4	5

👩	🍎	🍌🥝
Value	3	3
MMS Value	3	

👨	🍎🍌	🥝
Value	8	5
MMS Value	5	



What is Known?

- Finding MMS value is NP-hard
  - PTAS for finding MMS value [W97]

Existence (MMS allocation)?

- $n = 2$: YES 

- $n > 2$: NO [PW14]

What is Known?

- Finding MMS value is NP-hard
  - PTAS for finding MMS value [W97]

Existence (MMS allocation)?

- $n = 2$: YES EXERCISE
- $n > 2$: NO [PW14]

- α-MMS allocation: $v_i(A_i) \geq \alpha . \mu_i$
  - $2/3$-MMS exists [PW14, AMNS17, BK17, KPW18, G.MT18]
  - $3/4$-MMS exists [GHSSY18]
  - $(3/4 + 1/(12n))$-MMS exists [G.T20]

Properties

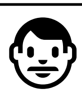
- Normalized valuations
  - Scale free: $v_{ij} \leftarrow c.v_{ij}, \forall j \in M$
  - $\sum_j v_{ij} = n \quad \Rightarrow \quad \mu_i \leq 1$

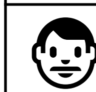
Properties

- Normalized valuations
  - Scale free: $v_{ij} \leftarrow c.v_{ij}, \forall j \in M$
  - $\sum_j v_{ij} = n \quad \Rightarrow \quad \mu_i \leq 1$
- Ordered Instance: We can assume that agents' order of preferences for items is same: $v_{i1} \geq v_{i2} \geq \cdots v_{im}, \forall i \in N$

Properties

- Normalized valuations
  - Scale free: $v_{ij} \leftarrow c.v_{ij}, \forall j \in M$
  - $\sum_j v_{ij} = n \quad \Rightarrow \quad \mu_i \leq 1$
- Ordered Instance: We can assume that agents' order of preferences for items is same: $v_{i1} \geq v_{i2} \geq \cdots v_{im}, \forall i \in N$

					
	3	1	2	5	4
	4	4	5	3	2

	1	**2**	**3**	**4**	**5**
	5	4	3	2	1
	5	4	4	3	2

Properties

- Normalized valuations
  - Scale free: $v_{ij} \leftarrow c.v_{ij}, \forall j \in M$
  - $\sum_j v_{ij} = n \quad \Rightarrow \quad \mu_i \leq 1$
- Ordered Instance: We can assume that agents' order of preferences for items is same: $v_{i1} \geq v_{i2} \geq \cdots v_{im}, \forall i \in N$
- Valid Reduction (α-MMS): If there exists $S \subseteq M$ and $i^* \in N$
  - $v_{i^*}(S) \geq \alpha.\mu_{i^*}^n(M)$
  - $\mu_i^{n-1}(M \setminus S) \geq \mu_i^n(M), \forall i \neq i^*$

  $\Rightarrow$ We can reduce the instance size!

Challenge

- Allocation of high-value items!
- If for all $i \in N$
  - $v_i(M) = n \Rightarrow \mu_i \leq 1$
  - $v_{ij} \leq \epsilon, \forall i, j$

Challenge

- Allocation of high-value items!
- If for all $i \in N$
  - $v_i(M) = n \Rightarrow \mu_i \leq 1$
  - $v_{ij} \leq \epsilon, \forall i, j$

Bag Filling Algorithm for $(1-\epsilon)$-MMS allocation:

Repeat until every agent is assigned a bag

- Start with an empty bag B
- Keep adding items to B until some agent i values it $\geq (1-\epsilon)$
- Assign B to i and remove them

Warm Up: 1/2-MMS Allocation

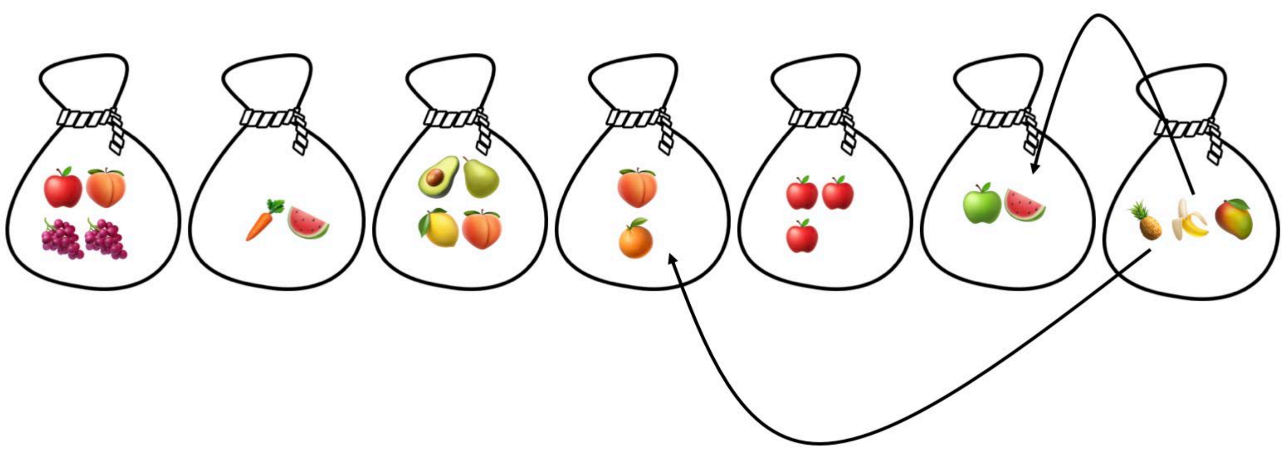
- Assume that μ_i is known for all i
  - Scale valuations so that $\mu_i = 1 \Rightarrow v_i(M) \geq n$
- If all $v_{ij} \leq 1/2$ then ?

1/2-MMS Allocation

- Assume that μ_i is known for all i
  - Scale valuations so that $\mu_i = 1 \Rightarrow v_i(M) \geq n$

Step 1: Valid Reductions
- If $v_{i1} \geq 1/2$ then assign item 1 to i

Step 2: Bag Filling

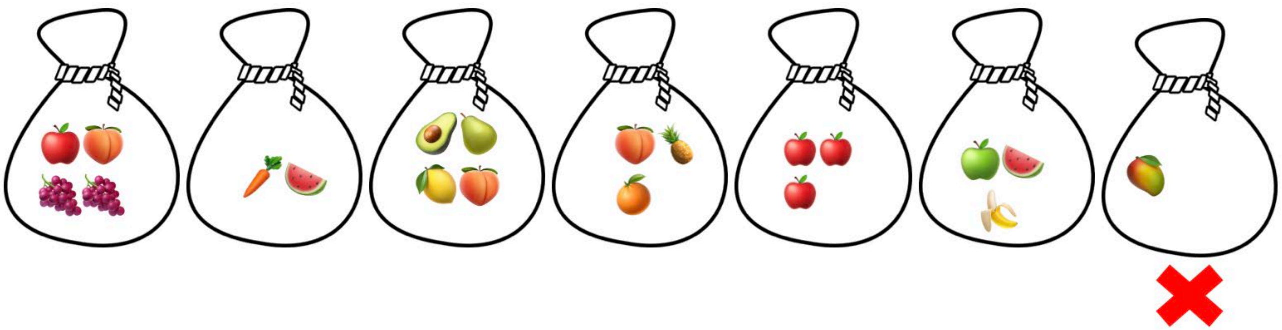


1/2-MMS Allocation

- Assume that μ_i is known for all i
  - Scale valuations so that $\mu_i = 1 \Rightarrow v_i(M) \geq n$

Step 1: Valid Reductions
- If $v_{i1} \geq 1/2$ then assign item 1 to i

Step 2: Bag Filling



1/2-MMS Allocation

- μ_i is not known

Step 0: Normalize Valuations: $\sum_j v_{ij} = n \Rightarrow \mu_i \leq 1$

Step 1: Valid Reductions

- If $v_{i1} \geq 1/2$ then assign item 1 to i
- After every valid reduction, normalize valuations

Step 2: Bag Filling

2/3-MMS Allocation [G.MT19]

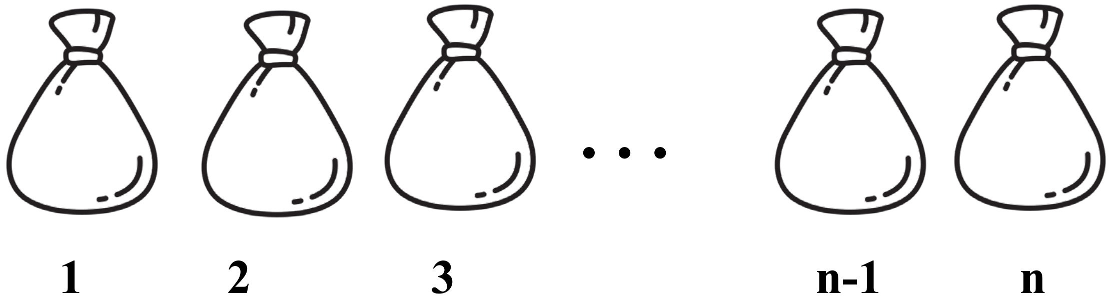
- Assume that μ_i is known for all i
  - Scale valuations so that $\mu_i = 1 \Rightarrow v_i(M) \geq n$
- If all $v_{ij} \leq 1/3$ then ?

Step 1: Valid Reductions

- If $v_{i1} \geq 2/3$ then assign item 1 to i
- If $v_{in} + v_{i(n+1)} \geq 2/3$ then assign $\{n, n+1\}$ to i

Step 2: Generalized Bag Filling

- Initialize n bags $\{B_1, \ldots B_n\}$ with $B_k = \{k\}, \forall k$



- Assume that μ_i is known for all i
  - Scale valuations so that $\mu_i = 1 \Rightarrow v_i(M) \geq n$

Step 1: Valid Reductions

- If $v_{i1} \geq 2/3$ then assign item 1 to i
- If $v_{in} + v_{i(n+1)} \geq 2/3$ then assign $\{n, n+1\}$ to i

Step 2: Generalized Bag Filling

- Initialize n bags $\{B_1, \ldots B_n\}$ with $B_k = \{k\}, \forall k$

1 2 3 … n-1 n

2/3-MMS Allocation [G.MT19]

- μ_i is not known

Step 0: Normalize Valuations: $\sum_j v_{ij} = n \Rightarrow \mu_i \leq 1$

Step 1: Valid Reductions

- If $v_{i1} \geq 2/3$ then assign item 1 to i
- If $v_{in} + v_{i(n+1)} \geq 2/3$ then assign $\{n, n+1\}$ to i
- After every valid reduction, normalize valuations

Step 2: Generalized Bag Filling

- Initialize n bags $\{B_1, \dots B_n\}$ with $B_k = \{k\}, \forall k$

Summary

Covered

- Additive Valuations:
  - Prop1 + PO (polynomial-time algorithm)
  - 2/3-MMS allocation (polynomial-time algorithm)

Not Covered

- More general valuations
  - MMS [GHSSY18]
- Groupwise-MMS [BBKN18]
- Chores
  - 11/9-MMS [HL19]

Major Open Questions (additive)

- c-MMS + PO: polynomial-time algorithm for a constant $c > 0$
- Existence of 4/5-MMS allocation? For 5 agents?

- [AMNS17] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. "Approximation algorithms for computing maximin share allocations". In: *ACM Trans. Algorithms 13.4 (2017)*
- [BBKN18] Siddharth Barman, Arpita Biswas, Sanath Kumar Krishnamurthy, and Y. Narahari. "Groupwise maximin fair allocation of indivisible goods". In: *AAAI 2018*
- [BK17] Siddharth Barman and Sanath Kumar Krishna Murthy. "Approximation algorithms for maximin fair division". In *EC 2017*
- [BK19] Siddharth Barman and Sanath Kumar Krishnamurthy. "On the Proximity of Markets with Integral Equilibria" In *AAAI 2019*
- [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
- [G.MT19] Jugal Garg, Peter McGlaughlin, and Setareh Taki. "Approximating Maximin Share Allocations". In: *SOSA@SODA 2019*

- [G.T20] Jugal Garg and Setareh Taki. "An Improved Approximation Algorithm for Maximin Shares". In: *EC 2020*
- [GHSSY18] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. "Fair allocation of indivisible goods: Improvement and generalization". In *EC 2018*
- [HL19] Xin Huang and Pinyan Lu. "An algorithmic framework for approximating maximin share allocation of chores". In: *arxiv:1907.04505*
- [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
- [KPW18] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. "Fair Enough: Guaranteeing Approximate Maximin Shares". In: *J. ACM 65.2 (2018), 8:1–8:27*
- [PW14] Ariel D Procaccia and Junxing Wang. "Fair enough: Guaranteeing approximate maximin shares". In *EC 2014*
- [W97] Gerhard J Woeginger. "A polynomial-time approximation scheme for maximizing the minimum machine completion time". In: *Operations Research Letters 20.4 (1997)*