Fair Division of Indivisible Items (Part II)

CS 580

Instructor: Ruta Mehta

Most slides are courtesy Prof. J. Garg
- $N$: set of $n$ 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
Proportionality

- A set $N$ of $n$ agents, a set $M$ of $m$ indivisible goods

- **Proportionality**: Allocation $A = (A_1, \ldots, A_n)$ is proportional if each agent gets at least $1/n$ share of all items:

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

Cut-and-choose?
Maximin Share (MMS) [B11]

Cut-and-choose.

- 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.
- $\mu_i := $ Maximum value of $i$’s least preferred bundle
Maximin Share (MMS) [B11]

Cut-and-choose.

- 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.
- $\mu_i := \text{Maximum value of } i\text{'s least preferred bundle}$

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

- MMS Allocation: $A$ is called MMS if $v_i(A_i) \geq \mu_i$, $\forall i$
- Additive valuations: $v_i(A_i) = \sum_{j \in A_i} v_{ij}$
### MMS value/partition/allocation

<table>
<thead>
<tr>
<th>Agent\Items</th>
<th>Item</th>
<th>Item</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>apple</td>
<td>banana</td>
<td>kiwi</td>
</tr>
<tr>
<td>🎨</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>🍎</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>MMS Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>MMS Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

---

- A: 🎨:🍎, 🍎
- B: 🍎, 🍎, 🍎, 🍎
MMS value/partition/allocation

Finding MMS value is NP-hard!
What is Known?

- PTAS for finding MMS value [W97]

Existence (MMS allocation)?

- $n = 2$ : yes
  - A PTAS to find $(1 - \epsilon)$-MMS allocation for any $\epsilon > 0$
- $n \geq 3$ : NO [PW14]
What is Known?

- PTAS for finding MMS value \([W97]\)

Existence (MMS allocation)?
- \(n = 2\) : yes
  \[\Rightarrow \text{A PTAS to find } (1 - \epsilon)\text{-MMS allocation for any } \epsilon > 0\]
- \(n \geq 3\) : NO \([PW14]\)

- \(\alpha\)-MMS allocation for \(\alpha \in [0,1]\): \(\nu_i(A_i) \geq \alpha \cdot \mu_i\)
  - \(2/3\)-MMS exists \([PW14, AMNS17, BK17, KPW18, GMT18]\)
  - \(3/4\)-MMS exists \([GHSSY18]\)
  - \((3/4 + 1/(12n))\)-MMS exists \([GT20]\)

\[
\frac{5}{4} - \text{MMS does not exist}
\]
Properties

- Normalized valuations
  - Scale free: \( v_{ij} \leftarrow c \cdot v_{ij}, \forall j \in M \)
  - \( \sum_j v_{ij} = n \Rightarrow \mu_i \leq 1 \)
  
\[ v_i(M) \]

\[ \sum_{i} v_i(A_i) \geq \ldots \geq v_i(A_n) = \mu_i > 1 \]

\[ \sum_{i} v_i(A_k) = v_i(M) > n! \]
Properties

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

- **Normalized valuations**
  - **Scale free**: \( v_{ij} \leftarrow c \cdot v_{ij} , \forall j \in M \)
  - \( \sum_j v_{ij} = n \Rightarrow \mu_i \leq 1 \)

- **Ordered Instance**: We can assume that agents’ order of preferences for items is same: \( v_{i1} \geq v_{i2} \geq \cdots v_{im}, \forall i \in N \)

<table>
<thead>
<tr>
<th></th>
<th>🍎</th>
<th>🍌</th>
<th>🍑</th>
<th>🍍</th>
<th>🍑</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧑</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>🧑</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧑</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>🧑</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Challenge

- Allocation of high-value items!
- If for all $i \in N$
  - $v_i(M) = n \Rightarrow \mu_i \leq 1$
  - $v_{ij} \leq \epsilon, \forall i, j \Rightarrow \max_{i,j} v_{ij} \leq \epsilon$
Claim: After round $k$, if $i$ remains then $v_i(\text{remaining goods}) \geq n - k$.

Bag Filling Algorithm:

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 both

\[
v_{ij} \leq \epsilon, \forall i,j \quad \text{for} \quad g_1 > g_2 > \cdots > g_m
\]
$v_{ij} \leq \epsilon, \forall i, j \quad \forall i \in \mathbb{N}$

Claim: After round $k$, if $i$ remains then $v_i(\text{remaining goods}) \geq n - k$.

"Claim: In every round, value $B$ be assigned set $B$ items for agent $i < 1$."

Bag Filling Algorithm:

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 both
\[ v_{ij} \leq \epsilon, \forall i, j \]

**Thm:** Every agent gets at least \((1 - \epsilon)\).

**Bag Filling Algorithm:**

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 both
Warm Up: 1/2-MMS Allocation

- If all $v_{ij} \leq 1/2$ then?
  - Done, using bag filling.

- What if some $v_{ij} > 1/2$?
Valid Reductions

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

- Ordered Instance: Agents’ order of preferences for items is same: $v_{i1} \geq v_{i2} \geq \ldots v_{im}, \forall i \in N$

- Valid Reduction ($\alpha$-MMS): If there exists $S \subseteq M$ and $i^* \in N$
  - $v_{i^*}(S) \geq \alpha.\mu_{i^*}^n(M)$
  - $\mu_i^{n-1}(M \setminus S) \geq \mu_i^n(M), \forall i \neq i^*$

Claim. Suppose agent $i \neq i^*$ gets $A_i$ in the be an $\alpha$-MMS allocation of $M \setminus S$ to agents $N \setminus \{i^*\}$, then $(A_1, \ldots, A_{i^*-1}, S, A_{i^*+1}, \ldots, A_n)$ is an $\alpha$-MMS allocation in the original instance.

\[ v_i(A_i) \geq \alpha \mu_i^{n-1}(M \setminus S) \geq \alpha.\mu_i^n(M) \]
1/2-MMS Allocation

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

Step 1: Valid Reductions

- If $v_{i^*1} \geq 1/2$ then assign item 1 to $i^*$
1/2-MMS Allocation

- Re-normalization

ζ = \frac{1}{2}

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

Step 1: Valid Reductions

- If \( v_{i^*1} \geq 1/2 \) then assign item 1 to \( i^* \). Remove good 1 and agent \( i^* \)
- After every valid reduction, normalize valuations

Step 2: Bag Filling

\( v_{ij} < \frac{1}{2} = \zeta \)
2/3-MMS Allocation [GMT19]

- If all $v_{ij} \leq 1/3$ then?

**Step 1: Valid Reductions**
- If $v_{i^*1} \geq 2/3$ then assign item 1 to $i^*$
2/3-MMS Allocation [GMT19]

**Step 1: Valid Reductions**

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

For agent $i \neq i^*$, let the MMS defining partition be $\mathcal{I} = \{I_1, \ldots, I_n\}$. Then move $A_k \setminus \{n, n+1\}$ to other bundles & remove $A_k$.

Partition $A_k \setminus \{n, n+1\}$ into $(n-1)$ bundles each with value $\geq u_i$. Why valid reduction?
2/3-MMS Allocation [GMT19]

**Step 1: Valid Reductions**

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

Case II: $n \in A_k$, with items $j_1 < j_2 \leq (n + 1)$. Then, swap items $j_1$ and $n$, and items $j_2$ and $(n + 1)$. This may only increase $v_i(A_k)$ & $v_i(A_l)$ because $v_i(j_1) \geq v_i(n)$ & $v_i(j_2) \geq v_i(n + 1)$. 

For agent $i \neq i^*$, let the MMS defining partition be $\exists A_d$, with items $j_1 < j_2 \leq (n + 1)$. Then, swap items $j_1$ and $n$, and items $j_2$ and $(n + 1)$. This may only increase $v_i(A_k)$ & $v_i(A_l)$ because $v_i(j_1) \geq v_i(n)$ & $v_i(j_2) \geq v_i(n + 1)$.
2/3-MMS Allocation [GMT19]

Step 1: Valid Reductions

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

Case II:

For agent $i \neq i^*$, let the MMS defining partition be $\exists A_d$, with items $j_1 < j_2 \leq (n + 1)$. Then, swap items $j_1$ and $n$, and items $j_2$ and $(n+1)$. Move remaining items of $A_d$ to other bundles and remove $A_d$. 
2/3-MMS Allocation [GMT19]

Step 1: Valid Reductions

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

For agent $i \neq i^*$, let the MMS defining partition be

Case II: $n \in A_k$ and $(n+1) \in A_l$

Again, value of none of the remaining bundles has decreased.

$\Rightarrow$ MMS value of agent $i$ has only increased in the reduced instance.
Step 1: Valid Reductions

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

Step 2: Generalized Bag Filling with $\epsilon = \frac{1}{3}$

- Initialize $n$ bags $\{B_1, ... B_n\}$ with $B_k = \{k\}, \forall k$.
- Assign items starting from $(n+1)$th to the first available bag, and give it to the first agent who shouts (values it at least $2/3 = (1 - \epsilon)$).

Claim. If agent $i^*$ is the first to shout, then for any agent $i \neq i^*$ the bag is of value at most 1.
2/3-MMS Allocation [GMT19]

(Re)normalization

Step 0: Normalized Valuations: \( \sum_j v_{ij} = n \Rightarrow \mu_i \leq 1 \)

Step 1: Valid Reductions

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

Step 2: Generalized Bag Filling with \( \epsilon = \frac{1}{3} \)

- Initialize \( n \) bags \( \{B_1, \ldots, B_n\} \) with \( B_k = \{k\}, \forall k \)
Chores
- **N**: set of *n* agents, 1, ..., *n*,
- **M**: set of *m* indivisible chores

**Agent** *i* has a disutility function \( d_i : 2^m \rightarrow \mathbb{R}_+ \) over subsets of items

- **Monotone**: the more the unhappier
- **Additive**: \( d_i(S) = \sum_{j \in S} d_{ij}, \) for any subset \( S \subseteq M \)
\begin{itemize}
  
  \item $N$: set of $n$ agents, 1 ,…, $n$,
  \item $M$: set of $m$ indivisible chores
  \item Agent $i$ has a \textit{disutility} function $d_i : 2^m \rightarrow \mathbb{R}_-$ over subsets of items
    \begin{itemize}
      \item Additive: $d_i(S) = \sum_{j \in S} d_{ij}$, for any subset $S \subseteq M$
    \end{itemize}
    Allocation $A = (A_1, \ldots, A_n)$

\end{itemize}

\textbf{EF1:} No agent envies another after removing one of her chores.

\[ \forall i, k \in N, \quad d_i(A_i \setminus c) \leq d_i(A_k), \quad \exists c \in A_i \]
EF1: Algorithms

Round Robin

1. Order agents arbitrarily.
2. Let them pick their best chore (least painful chore), one-at-a-time, in that order.

Observations:
- If agent \( k \) picks the last chore, then agent \( (k + 1) \) does not envy anyone. Why?
EF1: Algorithms

Envy-cycle-elimination

1. \( A = (\emptyset, \ldots, \emptyset) \)

2. While there are unassigned chores
   2. Give an unassigned chore to ..... ??

Observations:

- Cycle elimination does not increase any agent’s disutility.
- Giving a chore to sink maintains EF1. Why?
MMS

- $N$: set of $n$ agents, 1, …, n,
- $M$: set of $m$ indivisible chores
- Agent $i$ has a disutility function $d_i : 2^m \to \mathbb{R}_-$ over subsets of items
  - Additive: $d_i(S) = \sum_{j \in S} d_{ij}$, for any subset $S \subseteq M$

- $\Pi :=$ Set of all partitions of items into $n$ bundles

**MMS value:** $\text{MMS}_i = \mu_i = \min_{A \in \Pi} \max_{A_k \in A} d_i(A_k)$

$\alpha$-MMS allocation for $\alpha \geq 1$: $\forall i, d_i(A_i) \leq \alpha \mu_i$

1-MMS allocation may not exist!
EF1 to $\alpha$-MMS

Claim. If $(A_1, \ldots, An)$ is EF1 then it is 2-MMS

Observations: $\mu_i \geq \frac{d_i(M)}{n}$ and $\mu_i \geq \max_{j \in M} d_{ij}$

Proof.
Summary

Covered

- Additive Valuations:
  - \( \frac{1}{2} \)-MMS allocation (poly-time algorithm)
  - \( \frac{2}{3} \)-MMS allocation (polynomial-time algorithm)

State-of-the-art

- \( \left( \frac{3}{4} + \right) \)-MMS allocation [GT20]
- 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?
References (Indivisible Case).


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


GMT19 Jugal Garg, Peter McGlaughlin, and Setareh Taki. “Approximating Maximin Share Allocations”. In: SODA@SODA 2019

GT20 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


PW14 Ariel D Procaccia and Junxing Wang. “Fair enough: Guaranteeing approximate maximin shares”. In EC 2014