Fair Division of Indivisible Items

Jugal Garg

21st Max Planck Advanced Course on the Foundations of Computer Science (ADFOCS)
August 24-28, 2020
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.
- $\mu_i :=$ Maximum value of $i$'s least preferred bundle.

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

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

- \( \alpha \)-MMS allocation: \( v_i(A_i) \geq \alpha \cdot \mu_i \)
  - 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: $v_{ij} \leftarrow c \cdot v_{ij}$, $\forall j \in M$
  - $\sum_j v_{ij} = n \Rightarrow \mu_i \leq 1$

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

- **Valid Reduction ($\alpha$-MMS):** If there exists $S \subseteq M$ and $i^* \in N$
  - $v_{i^*}(S) \geq \alpha \cdot \mu_{i^*}^n(M)$
  - $\mu_{i}^{n-1}(M \setminus S) \geq \mu_i^n(M), \forall i \neq i^*$
  - $\Rightarrow$ We can reduce the instance size!
Challenge

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

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
1/2-MMS Allocation

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

**Step 1: Valid Reductions**
- If $v_{i_1} \geq 1/2$ then assign item 1 to $i$

**Step 2: Bag Filling**
1/2-MMS Allocation

- **Assume** that $\mu_i$ 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

- $\mu_i$ 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 $\mu_i$ 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 $\mu_i$ 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 $B_k = \{k\}, \forall k$
2/3-MMS Allocation [G.MT19]

- $\mu_i$ 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 $\{B_1, \ldots, B_n\}$ with $B_k = \{k\}, \forall k$
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)**

<table>
<thead>
<tr>
<th></th>
<th>EF1</th>
<th>EFX</th>
<th>Lecture 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMS</td>
<td>Prop1</td>
<td></td>
<td>Lecture 4</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Guarantees</td>
<td>Lecture 5</td>
</tr>
</tbody>
</table>

J. Garg (ADFOCS 2020)
Objectives

- Maximize the sum of valuations (Utilitarian Welfare):
  \[ UW(A) = \sum_i v_i(A_i) \]
Objectives

- Maximize the sum of valuations (Utilitarian Welfare):
  \[ UW(A) = \sum_i v_i(A_i) \]

- Maximize the minimum of valuations (Max-Min-Fairness, Egalitarian Welfare):
  \[ EW(A) = \min_i v_i(A_i) \]
Objectives

- Maximize the sum of valuations (Utilitarian Welfare):
  \[ UW(A) = \sum_{i} v_i(A_i) \]

- Maximize the minimum of valuations (Max-Min-Fairness, Egalitarian Welfare):
  \[ EW(A) = \min_{i} v_i(A_i) \]

- Maximize the geometric mean of valuations (\approx Efficiency + Fairness, Maximum Nash Welfare):
  \[ NW(A) = \left( \prod_{i \in A} v_i(A_i) \right)^{1/n} \] Scale-free

J. Garg (ADFOCS 2020)
Maximum Nash Welfare (MNW)

- **Maximum Nash welfare (MNW):** An allocation $A$ that maximizes the Nash welfare among all feasible allocations i.e.,

$$A^* = \arg \max_A (\prod_i v_i(A_i))^{1/n}$$

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

- **Divisible Items:** $\text{MNW} \equiv \text{CEEI} \Rightarrow \text{Envy-free + Prop + PO + …}$

- **Indivisible Items:** $\text{MNW} \Rightarrow \text{EF1 + PO + } \Omega(\frac{1}{\sqrt{n}})-\text{MMS [CKMPSW16]}$

  - Existence of EF1 + PO allocation
MNW (additive)

- APX-hard [Lee17]; 1.069-hardness [G.HM18]

**Approximation:**

- $\rho$-approximate MNW allocation $A$: $\rho \cdot NW(A) \geq MNW$
  - $2$ [CG15, CDGJMVY17], $e$ [AOSS17]
  - $1.45$ [BKV18] (pEF1 approach)

- Fairness Guarantees
  - Prop1 + PO + $\frac{1}{2n}$-MMS + 2-MNW [G.M19]

*Close the gaps!*
MNW: Generalizations

- Non-symmetric Agents (different entitlements/weights)
  - Weighted envy-free, weighted proportionality
  - MNW (weighted geometric mean)

- Beyond Additive Valuations

\[
\text{Additive} \subset \text{SC} \subset \text{OXS} \subset \text{Rado Budget additive} \subset \text{Submodular} \subset \text{Subadditive}
\]
MNW: Generalizations

- **Non-symmetric Agents (different entitlements/weights)**
  - Weighted envy-free, weighted proportionality
  - MNW (weighted geometric mean)

- **Beyond Additive Valuations**

  \[
  \text{Additive} \subset \text{SC} \subset \text{OXS} \subset \text{Rado} \subset \text{Submodular} \subset \text{Subadditive}
  \]
The **non-symmetric** MNW Problem

- Non-symmetric MNW was proposed in [HS72, K77] and has been extensively studied and used in many applications
  - Agent $i$ has a weight of $w_i$

$$NW(A) = \left( \prod_{i} v_i(A_i) \right)^{w_i/\sum_i w_i}$$

- MNW = $\arg \max_A NW(A)$

- $\rho$-approximate MNW allocation $A$: $\rho \cdot NW(A) \geq MNW$
Example (additive)

\[ w_i \]

\[
\begin{align*}
1 & \quad [10, 10, 1] \\
1 & \quad [1, 2, 1]
\end{align*}
\]

\[ MNW = NW(A) = (10^1 \cdot 3^1)^{1/2} \]
Example (additive)

\[ NW(A) = (10^2 \cdot 3^1)^{1/3} \]

\[ w_i \]

2 \ [10, 10, 1]  

1 \ [1, 2, 1]
Example (additive)

\[ NW(A) = (10^2 \cdot 3^1)^{1/3} < (20^2 \cdot 1^1)^{1/3} = NW(A') = MNW \]
### MNW Approximations: Additive

<table>
<thead>
<tr>
<th></th>
<th>Lower bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Symmetric</strong></td>
<td>1.069</td>
<td>1.45</td>
</tr>
<tr>
<td><strong>Non-symmetric</strong></td>
<td>1.069</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

$n$: # of agents

Constant factor? sublinear?
Matching \((m = n)\)

\[
NW(A) = \left( \prod_i v_i(A_i) \right)^{w_i / \sum_i w_i}
\]

\[
MNW = \max_A NW(A) \equiv \max_A \sum_i w_i \log v_i(A_i)
\]

**Claim:** If \(m = n\), then max-weight matching outputs MNW
$m > n$

- How good is max-weight matching?

- Issue: Allocation of high-value items!

$\NW(A^*) \approx m$

$\NW(A) \approx \sqrt{2m}$
Round Robin Procedure

- $H_i$: Set of $n$ highest-valued items for agent $i$
- $u_i = v_i(M \setminus H_i)$
- Guarantee?
- $H_i$: Set of $n$ highest-valued items for agent $i$
- $u_i = v_i(M \setminus H_i)$
- Round-Robin guarantees $\geq \frac{u_i}{n}$

**MNW allocation $A^*$:**
- $g_i^*$: highest-valued item in $A_i^*$
- $v_i(A_i^*) \leq nv_i(g_i^*) + u_i$
  \[ \leq n \left( v_i(g_i^*) + \frac{u_i}{n} \right) \]

- If $v_i(A_i) \geq v_i(g_i^*) + \frac{u_i}{n}$, then $A$ is $O(n)$-approximation!
Matching + Round-Robin [G.KK20]

- $H_i$: Set of $2n$ highest-valued items for agent $i$
- $u_i = v_i(M \setminus H_i)$
- Allocate one item to each agent using max-weight matching with weights $w_i \log(v_i(g) + \frac{u_i}{n})$: $y_i^*$ is allocated to $i$
- $A \leftarrow$ Allocate remaining items using Round Robin
- $H_i$: Set of $2n$ highest-valued items for agent $i$
- $u_i = v_i(M \setminus H_i)$
- Allocate one item to each agent using max-weight matching with weights $w_i \log(v_i(g) + \frac{u_i}{n})$: $y_i^*$ is allocated to $i$
- $A \leftarrow$ Allocate remaining items using Round Robin

- $g_i^*$: highest-valued item in $A_i^*$
- $v_i(A_i^*) \leq 2nv_i(g_i^*) + u_i \implies \text{MNW} \leq 2n \left( \prod_i \left( v_i(g_i^*) + \frac{u_i}{n} \right)^{w_i} \right)^{1/\sum_i w_i}$
- $v_i(A_i) \geq v_i(y_i^*) + \frac{u_i}{n}$

\[ \implies \text{NW}(A) \geq \left( \prod_i \left( v_i(y_i^*) + \frac{u_i}{n} \right)^{w_i} \right)^{\frac{1}{\Sigma_i w_i}} \geq \left( \prod_i \left( v_i(g_i^*) + \frac{u_i}{n} \right)^{w_i} \right)^{\frac{1}{\Sigma_i w_i}} \]

**Theorem** [G.KK20]: $A$ is $2n$-MNW + EF1
Generalizations

- Non-symmetric Agents (different entitlements/weights)
  - Weighted envy-free, weighted proportionality
  - MNW (weighted geometric mean)

- Beyond Additive

\[
\text{Additive} \subset \text{SC} \subset \text{OXS} \subset \text{Rado Additive} \subset \text{Submodular} \subset \text{Subadditive}
\]

**non-negative monotone**: \( v(S) \leq v(T), \ S \subseteq T \)

**Subadditive**: \( v(A \cup B) \leq v(A) + v(B), \ \forall A, B \)
Additive valuations are restrictive

100
Additive valuations are restrictive

100

100

100
Additive valuations are restrictive

\[ 100 + 100 \neq 125 \]
### MNW Approximations: Symmetric Agents

Additive $\subseteq$ SC $\subseteq$ OXS $\subseteq$ Rado

- Budget additive
- Separable concave

<table>
<thead>
<tr>
<th>Valuation</th>
<th>Lower bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>1.069</td>
<td>1.45</td>
</tr>
<tr>
<td>Budget additive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Separable concave</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OXS</td>
<td>1.069</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Rado</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submodular</td>
<td>1.58</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Subadditive</td>
<td>$O(n^{1-\epsilon})$</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

$n$: # of agents
### MNW Approximations: Non-symmetric Agents

Additive $\subset$ SC $\subset$ OXS $\subset$ Rado  
Budget additive $\subset$ Submodular $\subset$ Subadditive

<table>
<thead>
<tr>
<th>Valuation</th>
<th>Lower bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>1.069</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Budget additive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Separable concave</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OXS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rado</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submodular</td>
<td>1.58</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Subadditive</td>
<td>$O(n^{1-\epsilon})$</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

$n$: # of agents

---

J. Garg (ADFOCS 2020)
Envy-free (EF) Allocation

Claim: An EF allocation $A$ is $O(n)$-approximation
1/2-EFX Allocation

- 1/2-EFX allocation $A$: $v_i(A_i) \geq \frac{1}{2} v_i(A_j \setminus g), \forall g \in A_j, \forall i, j$

**Claim:** If $|A_i| \geq 2$, $\forall i$, then $A$ is $O(n)$-approximation
\( O(n) \) Algorithm [CG.M20]

- \( H_i \): Set of \( n \) highest-valued items for agent \( i \)
- Allocate one item per agent using max-weight matching with weights \( w_i \log(v_i(g) + \frac{v_i(M \setminus H_i)}{n}) \) : \( y_i^* \) is allocated to \( i \)
- \( A \leftarrow \) Allocate remaining items using \( \frac{1}{2}\)-EFX algorithm

**Claim:** \( A \) is \( O(n) \)-MNW and \( \frac{1}{2}\)-EFX allocation
**Claim:** $A$ is $O(n)$-MNW

**Proof (sketch):**

- $Y \leftarrow \bigcup_i y_i^*; \quad g_i^*: \text{highest-valued item in MNW allocation } A_i^*$

- $v_i(A_i^*) \leq n v_i(g_i^*) + v_i(M \setminus H_i) = n \left( v_i(g_i^*) + \frac{v_i(M \setminus H_i)}{n} \right)$

$$\Rightarrow \quad \text{MNW} \leq n \left( \prod_i \left( v_i(g_i^*) + \frac{v_i(M \setminus H_i)}{n} \right)^{w_i} \right)^{1/ \sum_i w_i}$$

- $v_i(A_i) \geq v_i(y_i^*)$

- $v_i(A_i) \geq \frac{v_i(M \setminus Y)}{4n} \geq \frac{v_i(M \setminus H_i) - n v_i(y_i^*)}{4n}$

- $v_i(A_i) \geq \frac{1}{8} \left( v_i(y_i^*) + \frac{v_i(M \setminus H_i)}{n} \right)$

$$\text{NW}(A) \geq \frac{1}{8} \left( \prod_i \left( v_i(y_i^*) + \frac{v_i(M \setminus H_i)}{n} \right)^{w_i} \right)^{\frac{1}{\sum_i w_i}} \geq \frac{1}{8} \left( \prod_i \left( v_i(g_i^*) + \frac{v_i(M \setminus H_i)}{n} \right)^{w_i} \right)^{\frac{1}{\sum_i w_i}}$$

J. Garg (ADFOCS 2020)
### MNW Approximations: Symmetric Agents

Additive $\subseteq$ SC $\subseteq$ OXS $\subseteq$ Rado $\subseteq$ Submodular $\subseteq$ Subadditive

<table>
<thead>
<tr>
<th>Valuation</th>
<th>Lower bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>1.069</td>
<td>1.45</td>
</tr>
<tr>
<td>Budget additive</td>
<td>1.069</td>
<td>1.45</td>
</tr>
<tr>
<td>Separable concave</td>
<td>1.069</td>
<td>1.45</td>
</tr>
<tr>
<td>OXS</td>
<td>1.069</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Rado</td>
<td>1.069</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Submodular</td>
<td>1.58</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

$n$: # of agents
MNW Approximations: Non-symmetric Agents

Additive ⊂ SC ⊂ OXS ⊂ Rado ⊂ Submodular ⊂ Subadditive

<table>
<thead>
<tr>
<th>Valuation</th>
<th>Lower bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>1.069</td>
<td>O(n)</td>
</tr>
<tr>
<td>Budget additive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Separable concave</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OXS</td>
<td>1.58</td>
<td>0(n)</td>
</tr>
<tr>
<td>Rado</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submodular</td>
<td></td>
<td>0(n)</td>
</tr>
</tbody>
</table>

\( n \): # of agents


