### Computational Social Choice

**Fair Allocation of Indivisible Items**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$:</td>
<td>20</td>
<td>50</td>
<td>35</td>
<td>45</td>
<td>15</td>
<td>8</td>
<td>55</td>
</tr>
<tr>
<td>$v_2$:</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
<td>90</td>
</tr>
<tr>
<td>$v_3$:</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
<td>19</td>
</tr>
</tbody>
</table>

Piotr Skowron
University of Warsaw
The Model

Input:

1. A set of objects \( \mathcal{O} = \{o_1, o_2, \ldots, o_m\} \).
2. A set of voters \( V = \{v_1, v_2, \ldots, v_n\} \).
3. For each voter \( v_i \) and each object \( o_j \) we are given a utility \( u_i(o_j) \), that measures how much \( v_i \) likes \( o_j \).
The Model

Input:

1. A set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$.
2. A set of voters $V = \{v_1, v_2, \ldots, v_n\}$.
3. For each voter $v_i$ and each object $o_j$ we are given a utility $u_i(o_j)$, that measures how much $v_i$ likes $o_j$.

<table>
<thead>
<tr>
<th>$v_1$</th>
<th>20</th>
<th>50</th>
<th>35</th>
<th>45</th>
<th>15</th>
<th>8</th>
<th>55</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_2$</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>$v_3$</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
<td>19</td>
<td>82</td>
</tr>
</tbody>
</table>
The Model

Input:

1. A set of objects \( \mathcal{O} = \{o_1, o_2, \ldots, o_m\} \).
2. A set of voters \( V = \{v_1, v_2, \ldots, v_n\} \).
3. For each voter \( v_i \) and each object \( o_j \) we are given a utility \( u_i(o_j) \), that measures how much \( v_i \) likes \( o_j \).

Output:

An allocation \( \{A_i\}_{i \in V} \), where \( A_i \) denotes the set of items allocated to \( v_i \)

<table>
<thead>
<tr>
<th></th>
<th>( v_1 )</th>
<th>( v_2 )</th>
<th>( v_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>20</td>
<td>15</td>
<td>58</td>
</tr>
<tr>
<td>House</td>
<td>50</td>
<td>30</td>
<td>23</td>
</tr>
<tr>
<td>Plane</td>
<td>35</td>
<td>22</td>
<td>75</td>
</tr>
<tr>
<td>Tree</td>
<td>45</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Cake</td>
<td>15</td>
<td>12</td>
<td>42</td>
</tr>
<tr>
<td>Shoes</td>
<td>8</td>
<td>17</td>
<td>26</td>
</tr>
<tr>
<td>Boots</td>
<td>55</td>
<td>90</td>
<td>19</td>
</tr>
<tr>
<td>Hat</td>
<td>14</td>
<td>26</td>
<td>82</td>
</tr>
</tbody>
</table>
The Model

Input:
1. A set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$.
2. A set of voters $V = \{v_1, v_2, \ldots, v_n\}$.
3. For each voter $v_i$ and each object $o_j$ we are given a utility $u_i(o_j)$, that measures how much $v_i$ likes $o_j$.

Output:
An allocation $\{A_i\}_{i \in V}$, where $A_i$ denotes the set of items allocated to $v_i$. 

\begin{tabular}{cccccccc}
$v_1$: & 20 & 50 & 35 & 45 & 15 & 8 & 55 & 14 \\
$v_2$: & 15 & 30 & 22 & 0 & 12 & 17 & 90 & 26 \\
$v_3$: & 58 & 23 & 75 & 17 & 42 & 26 & 19 & 82 \\
\end{tabular}
The Model

Input:

1. A set of objects $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$.
2. A set of voters $V = \{v_1, v_2, \ldots, v_n\}$.
3. For each voter $v_i$ and each object $o_j$ we are given a utility $u_i(o_j)$, that measures how much $v_i$ likes $o_j$.

Output:

An allocation $\{A_i\}_{i \in V}$, where $A_i$ denotes the set of items allocated to $v_i$.

We want the allocation to be complete (that is each object should be allocated to someone).
The Model

Utilities are additive:

\[ \begin{align*}
  v_1: & \quad 20 \quad 50 \quad 35 \quad 45 \quad 15 \quad 8 \quad 55 \quad 14 \\
  v_2: & \quad 15 \quad 30 \quad 22 \quad 0 \quad 12 \quad 17 \quad 90 \quad 26 \\
  v_3: & \quad 58 \quad 23 \quad 75 \quad 17 \quad 42 \quad 26 \quad 19 \quad 82 
\end{align*} \]
The Model

Utilities are additive:

\[ u_1\left(\begin{array}{c}
\text{House} \\
\text{Ski} \\
\text{Tree}
\end{array}\right) = 55 + 45 + 15 = 115 \]

\[ u_2\left(\begin{array}{c}
\text{Office Building} \\
\text{Airplane} \\
\text{Trunk}
\end{array}\right) = 30 + 22 + 17 = 69 \]

\[ u_3\left(\begin{array}{c}
\text{Boat} \\
\text{Car}
\end{array}\right) = 82 + 58 = 140 \]
The Model

Utilities are additive:

\[ u_1 \left( \begin{array}{c}
\text{house} \\
\text{room} \\
\text{ocean}
\end{array} \right) = 55 + 45 + 15 = 115 \]

\[ u_2 \left( \begin{array}{c}
\text{building} \\
\text{airplane} \\
\text{suitcase}
\end{array} \right) = 30 + 22 + 17 = 69 \]

\[ u_3 \left( \begin{array}{c}
\text{boat} \\
\text{car}
\end{array} \right) = 82 + 58 = 140 \]

For each \( S \subseteq \emptyset \) we write:

\[ u_i(S) = \sum_{o \in S} u_i(o) \]
Axioms for Fairness

Proportional allocation: We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n \).
Axioms for Fairness

**Proportional allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n \).

**Envy-free allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).
Axioms for Fairness

**Proportional allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n. \)

**Envy-free allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j). \)

**Proposition:** An envy-free allocation satisfies proportional fair share.
## Axioms for Fairness

**Proportional allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n \).

**Envy-free allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

**Proposition:** An envy-free allocation satisfies proportional fair share.

**Proof:**

Take an envy-free allocation \( \{A_i\}_{i \in V} \), and fix a voter \( v_i \).
Axioms for Fairness

**Proportional allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n \).

**Envy-free allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

**Proposition:** An envy-free allocation satisfies proportional fair share.

**Proof:**

Take an envy-free allocation \( \{A_i\}_{i \in V} \), and fix a voter \( v_i \).
We have that: \( u_i(A_i) \geq u_i(A_j) \)
Axioms for Fairness

**Proposition:** An envy-free allocation satisfies proportional fair share.

Proof:

Take an envy-free allocation \( \{A_i\}_{i \in V} \), and fix a voter \( v_i \).
We have that:
\[
\begin{align*}
    u_i(A_i) &\geq u_i(A_1) \\
    u_i(A_i) &\geq u_i(A_2)
\end{align*}
\]
Axioms for Fairness

**Proportional allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n \).

**Envy-free allocation:** We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

**Proposition:** An envy-free allocation satisfies proportional fair share.

**Proof:**

Take an envy-free allocation \( \{A_i\}_{i \in V} \), and fix a voter \( v_i \).

We have that:

\[
\begin{align*}
u_i(A_i) & \geq u_i(A_1) \\
u_i(A_i) & \geq u_i(A_2) \\
& \vdots \\
u_i(A_i) & \geq u_i(A_n)
\end{align*}
\]
Axioms for Fairness

Proportional allocation: We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n. \)

Envy-free allocation: We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j). \)

Proposition: An envy-free allocation satisfies proportional fair share.

Proof:

Take an envy-free allocation \( \{A_i\}_{i \in V} \), and fix a voter \( v_i \).
We have that: \( u_i(A_i) \geq u_i(A_1) \)
\( u_i(A_i) \geq u_i(A_2) \)
\( \vdots \)
\( u_i(A_i) \geq u_i(A_n) \)
After summing up:
\[ nu_i(A_i) \geq \sum_{j=1}^{n} u_i(A_j) = u_i(\emptyset); \]
Axioms for Fairness

Proportional allocation: We say that an allocation \( \{A_i\}_{i \in V} \) is proportional if for each voter \( v_i \) we have that \( u_i(A_i) \geq u_i(\emptyset)/n \).

Envy-free allocation: We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

Proposition: An envy-free allocation satisfies proportional fair share.

Proof:

Take an envy-free allocation \( \{A_i\}_{i \in V} \), and fix a voter \( v_i \).

We have that:
\[
\begin{align*}
  u_i(A_i) &\geq u_i(A_1) \\
  u_i(A_i) &\geq u_i(A_2) \\
  &\vdots \\
  u_i(A_i) &\geq u_i(A_n)
\end{align*}
\]

After summing up:
\[
n u_i(A_i) \geq \sum_{j=1}^{n} u_i(A_j) = u_i(\emptyset); \text{ thus: } u_i(A_i) \geq u_i(\emptyset)/n.
\]
Envy-free allocations

Envy-free allocation (EF): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

Does an envy-free allocation always exist?
Envy-free allocations

Envy-free allocation (EF): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

Does an envy-free allocation always exist? **Unfortunately no...**
Envy-free allocations

Envy-free allocation (EF): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

Does an envy-free allocation always exist? **Unfortunately no...**

Example:
- Two voters \( v_1 \) and \( v_2 \) and one object \( o \); the utilities are: \( u_1(o) = u_2(o) = 1 \).
- If we allocate the object to one of the voters, then the other voter will envy.
Envy-free allocations

Envy-free allocation (EF): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

Does an envy-free allocation always exist? \textbf{Unfortunately no...}

Example:
- Two voters \( v_1 \) and \( v_2 \) and one object \( o \); the utilities are: \( u_1(o) = u_2(o) = 1 \).
- If we allocate the object to one of the voters, then the other voter will envy.

This example also shows that a proportional allocation does not always exist.
Envy-free allocations

Envy-free allocation (EF): We say that an allocation \( \{ A_i \}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

Does an envy-free allocation always exist? Unfortunately no...

Example:
- Two voters \( v_1 \) and \( v_2 \) and one object \( o \); the utilities are: \( u_1(o) = u_2(o) = 1 \).
- If we allocate the object to one of the voters, then the other voter will envy.

This example also shows that a proportional allocation does not always exist.
So... is there a fairness axiom that would be satisfiable?
Envy-free allocations

Envy-free allocation (EF): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free if for each \( v_i \) and \( v_j \) we have that \( u_i(A_i) \geq u_i(A_j) \).

Does an envy-free allocation always exist? **Unfortunately no...**

Example:
- Two voters \( v_1 \) and \( v_2 \) and one object \( o \); the utilities are: \( u_1(o) = u_2(o) = 1 \).
- If we allocate the object to one of the voters, then the other voter will envy.

This example also shows that a proportional allocation does not always exist.

So... is there a fairness axiom that would be satisfiable?

Envy-freeness up to one good (EF1): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free up to one good if for each \( v_i, v_j \in V \) either \( A_j = \emptyset \) or there exists an object \( o \in A_j \) such that \( u_i(A_i) \geq u_i(A_j \setminus \{o\}) \).
Does an EF1 allocation always exist?

**Envy-freeness up to one good (EF1):** We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free up to one good if for each \( v_i, v_j \in V \) either \( A_j = \emptyset \) or there exists an object \( o \in A_j \) such that \( u_i(A_i) \geq u_i(A_j \setminus \{o\}) \).
EF1 allocations

Envy-freeness up to one good (EF1): We say that an allocation $\{A_i\}_{i \in V}$ is envy-free up to one good if for each $v_i, v_j \in V$ either $A_j = \emptyset$ or there exists an object $o \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{o\})$.

Does an EF1 allocation always exist? **YES!!**
EF1 allocations

Envy-freeness up to one good (EF1): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free up to one good if for each \( v_i, v_j \in V \) either \( A_j = \emptyset \) or there exists an object \( o \in A_j \) such that \( u_i(A_i) \geq u_i(A_j \setminus \{o\}) \).

Does an EF1 allocation always exist? **YES!!**

Round Robin Algorithm:
1. Voters take objects in a cyclic order.
2. In each turn, a voter picks the object she likes the most among those that are not yet picked.
### Round Robin Algorithm

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$:</td>
<td>20</td>
<td>50</td>
<td>35</td>
<td>45</td>
<td>15</td>
<td>8</td>
<td>55</td>
</tr>
<tr>
<td>$v_2$:</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
<td>90</td>
</tr>
<tr>
<td>$v_3$:</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
<td>19</td>
</tr>
</tbody>
</table>
# Round Robin Algorithm

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| $v_1$: | 20 | 50 | 35 | 45 | 15 | 8 | 55 | 14 |
| $v_2$: | 15 | 30 | 22 | 0 | 12 | 17 | 90 | 26 |
| $v_3$: | 58 | 23 | 75 | 17 | 42 | 26 | 19 | 82 |

$v_1$: [House]

$v_2$: 

$v_3$: 
Round Robin Algorithm

\[ v_1: \quad 20 \quad 35 \quad 45 \quad 15 \quad 8 \quad 55 \quad 14 \]
\[ v_2: \quad 15 \quad 30 \quad 22 \quad 0 \quad 12 \quad 17 \quad 90 \quad 26 \]
\[ v_3: \quad 58 \quad 23 \quad 75 \quad 17 \quad 42 \quad 26 \quad 19 \quad 82 \]
Round Robin Algorithm

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$: 20</td>
<td>50</td>
<td>35</td>
<td>45</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>$v_2$: 15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>$v_3$: 58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
</tr>
</tbody>
</table>
Round Robin Algorithm

<table>
<thead>
<tr>
<th></th>
<th>v_1</th>
<th>v_2</th>
<th>v_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>v_1</td>
<td>20</td>
<td>50</td>
<td>35</td>
</tr>
<tr>
<td>v_2</td>
<td>20</td>
<td>50</td>
<td>35</td>
</tr>
<tr>
<td>v_3</td>
<td>20</td>
<td>50</td>
<td>35</td>
</tr>
</tbody>
</table>

Diagram:
- v_1: Car, Building, Airplane, Man, Tree, Treasure Chest, House, Boat
- v_2: Car, Building, Airplane, Man, Tree, Treasure Chest, House, Boat
- v_3: Car, Building, Airplane, Man, Tree, Treasure Chest, House, Boat
Round Robin Algorithm

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>50</td>
<td>35</td>
<td>45</td>
<td>15</td>
<td>8</td>
<td>55</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
<td>19</td>
<td>82</td>
</tr>
</tbody>
</table>
### Round Robin Algorithm

<table>
<thead>
<tr>
<th>$v_1$</th>
<th>20</th>
<th>50</th>
<th>35</th>
<th>45</th>
<th>15</th>
<th>8</th>
<th>55</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_2$</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>$v_3$</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
<td>19</td>
<td>82</td>
</tr>
</tbody>
</table>
## Round Robin Algorithm

<table>
<thead>
<tr>
<th></th>
<th>v₁</th>
<th>v₂</th>
<th>v₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>v₁</td>
<td>20</td>
<td>50</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>v₂</td>
<td>15</td>
<td>30</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>v₃</td>
<td>58</td>
<td>23</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>42</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>82</td>
<td></td>
</tr>
</tbody>
</table>
Round Robin Algorithm

\[ v_1: \begin{array}{cccccccc}
20 & 50 & 35 & 45 & 15 & 8 & 55 & 14 \\
\end{array} \]

\[ v_2: \begin{array}{cccccccc}
15 & 30 & 22 & 0 & 12 & 17 & 90 & 26 \\
\end{array} \]

\[ v_3: \begin{array}{cccccccc}
58 & 23 & 75 & 17 & 42 & 26 & 19 & 82 \\
\end{array} \]
Properties of the Round Robin Algorithm

**Proposition:** Round Robin satisfies EF1.
Properties of the Round Robin Algorithm

**Proposition:** Round Robin satisfies EF1.

Proof:
Consider two voters, \( v_i \) and \( v_j \), that have been allocated bundles \( A_i \) and \( A_j \).
Properties of the Round Robin Algorithm

**Proposition:** Round Robin satisfies EF1.

Proof:
Consider two voters, $v_i$ and $v_j$, that have been allocated bundles $A_i$ and $A_j$.

Let $o_{i,1}$ be the first object selected by $v_i$, $o_{i,2}$ be the second one, etc.
Properties of the Round Robin Algorithm

Proposition: Round Robin satisfies EF1.

Proof:
Consider two voters, \( v_i \) and \( v_j \), that have been allocated bundles \( A_i \) and \( A_j \).

Let \( o_{i,1} \) be the first object selected by \( v_i \), \( o_{i,2} \) be the second one, etc.
Analogously, we define \( o_{j,1}, o_{j,2}, o_{j,3} \ldots \), as the first, second, third, ... object selected by \( v_j \).
Properties of the Round Robin Algorithm

Proposition: Round Robin satisfies EF1.

Proof:
Consider two voters, \( v_i \) and \( v_j \), that have been allocated bundles \( A_i \) and \( A_j \).

Let \( o_{i,1} \) be the first object selected by \( v_i \), \( o_{i,2} \) be the second one, etc. Analogously, we define \( o_{j,1}, o_{j,2}, o_{j,3}, \ldots \), as the first, second, third, ... object selected by \( v_j \).

We know that \( u_j(o_{j,1}) \geq u_j(o_{i,2}), u_j(o_{j,2}) \geq u_j(o_{i,3}), \text{ etc.} \)
Properties of the Round Robin Algorithm

**Proposition:** Round Robin satisfies EF1.

**Proof:**

Consider two voters, \( v_i \) and \( v_j \), that have been allocated bundles \( A_i \) and \( A_j \).

Let \( o_{i,1} \) be the first object selected by \( v_i \), \( o_{i,2} \) be the second one, etc. Analogously, we define \( o_{j,1}, o_{j,2}, o_{j,3} \ldots \), as the first, second, third, ... object selected by \( v_j \).

We know that \( u_j(o_{j,1}) \geq u_j(o_{i,2}), u_j(o_{j,2}) \geq u_j(o_{i,3}) \), etc.

Thus, after summing up: \( u_j(A_j) \geq u_j(A_i \setminus \{o_{i,1}\}) \).
Properties of the Round Robin Algorithm

Proposition: Round Robin satisfies EF1.

Pareto optimality (PO): An allocation $\{A_i\}_{i \in V}$ is dominated by $\{B_i\}_{i \in V}$ if:
1. for each voter $v_i \in V$ we have that $u_i(B_i) \geq u_i(A_i)$,
2. there exists $v_i \in V$ such that $u_i(B_i) > u_i(A_i)$.

We say that an allocation $\{A_i\}_{i \in V}$ is Pareto optimal if it is not dominated by any other allocation.
Properties of the Round Robin Algorithm

**Proposition:** Round Robin satisfies EF1.

**Pareto optimality (PO):** An allocation \( \{A_i\}_{i \in V} \) is dominated by \( \{B_i\}_{i \in V} \) if:
1. for each voter \( v_i \in V \) we have that \( u_i(B_i) \geq u_i(A_i) \),
2. there exists \( v_i \in V \) such that \( u_i(B_i) > u_i(A_i) \).

We say that an allocation \( \{A_i\}_{i \in V} \) is Pareto optimal if it is not dominated by any other allocation.
Properties of the Round Robin Algorithm

Proposition: Round Robin satisfies EF1.

Pareto optimality (PO): An allocation $\{A_i\}_{i \in V}$ is dominated by $\{B_i\}_{i \in V}$ if:
1. for each voter $v_i \in V$ we have that $u_i(B_i) \geq u_i(A_i)$,
2. there exists $v_i \in V$ such that $u_i(B_i) > u_i(A_i)$.

We say that an allocation $\{A_i\}_{i \in V}$ is Pareto optimal if it is not dominated by any other allocation.

Round Robin allocation might not be PO!
Proposition: Round Robin satisfies EF1.

Pareto optimality (PO): An allocation \( \{A_i\}_{i \in V} \) is dominated by \( \{B_i\}_{i \in V} \) if:
1. for each voter \( v_i \in V \) we have that \( u_i(B_i) \geq u_i(A_i) \),
2. there exists \( v_i \in V \) such that \( u_i(B_i) > u_i(A_i) \).

We say that an allocation \( \{A_i\}_{i \in V} \) is Pareto optimal if it is not dominated by any other allocation.

Round Robin allocation might not be PO!
Properties of the Round Robin Algorithm

**Proposition:** Round Robin satisfies EF1.

**Pareto optimality (PO):** An allocation \( \{A_i\}_{i \in V} \) is dominated by \( \{B_i\}_{i \in V} \) if:
1. for each voter \( v_i \in V \) we have that \( u_i(B_i) \geq u_i(A_i) \),
2. there exists \( v_i \in V \) such that \( u_i(B_i) > u_i(A_i) \).

We say that an allocation \( \{A_i\}_{i \in V} \) is Pareto optimal if it is not dominated by any other allocation.

Round Robin allocation might not be PO!
Is there an allocation that is EF1 and PO?
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \text{ (If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i).
\]

(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_1 ):</td>
<td>20</td>
<td>50</td>
<td>35</td>
<td>45</td>
<td>15</td>
<td>8</td>
<td>55</td>
<td>14</td>
</tr>
<tr>
<td>( v_2 ):</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>( v_3 ):</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
<td>19</td>
<td>82</td>
</tr>
</tbody>
</table>
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \text{ (If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

<table>
<thead>
<tr>
<th>Voter</th>
<th>City</th>
<th>Airplane</th>
<th>Tree</th>
<th>Carriage</th>
<th>House</th>
<th>Boat</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_1 )</td>
<td>20</td>
<td>50</td>
<td>35</td>
<td>45</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>( v_2 )</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>( v_3 )</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
</tr>
</tbody>
</table>

\( A: \) \( v_2: \) | \( B: \) \( v_2: \)
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \quad \text{(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

\[
\begin{align*}
\prod_{i \in V} u_i(A_i) &= (55 + 45 + 15) \cdot \left(30 + 22 + 17\right) \cdot (82 + 58) \\
&= 115 \cdot 79 \cdot 140
\end{align*}
\]
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \text{ (If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

\(\begin{array}{cccccccc}
\nu_1: & 20 & 50 & 35 & 45 & 15 & 8 & 55 & 14 \\
\nu_2: & 15 & 30 & 22 & 0 & 12 & 17 & 90 & 26 \\
\nu_3: & 58 & 23 & 75 & 17 & 42 & 26 & 19 & 82 \\
\end{array}\)

\(\begin{array}{c}
A: \nu_2: \\
\nu_1: \text{ House} & \text{ Tree} & \text{ Sun} \\
\nu_2: \text{ Building} & \text{ Airplane} & \text{ Suitcase} & \text{ House} \\
\nu_3: \text{ Boat} & \text{ Car} \\
\end{array}\)

\[
\prod_{i \in V} u_i(A_i) = (55 + 45 + 15) \cdot (30 + 22 + 17) \cdot (82 + 58)
\]

\(\begin{array}{c}
B: \nu_2: \\
\nu_1: \text{ Airplane} & \text{ Sun} & \text{ Tree} \\
\nu_2: \text{ Suitcase} & \text{ House} \\
\nu_3: \text{ Boat} & \text{ Car} \\
\end{array}\)

\[
\prod_{i \in V} u_i(B_i) = (45 + 15 + 35 + 50) \cdot (17 + 90) \cdot (82 + 58)
\]
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \quad \text{(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

<table>
<thead>
<tr>
<th>( v_1 )</th>
<th>20</th>
<th>50</th>
<th>35</th>
<th>45</th>
<th>15</th>
<th>8</th>
<th>55</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_2 )</td>
<td>15</td>
<td>30</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>17</td>
<td>90</td>
<td>26</td>
</tr>
<tr>
<td>( v_3 )</td>
<td>58</td>
<td>23</td>
<td>75</td>
<td>17</td>
<td>42</td>
<td>26</td>
<td>19</td>
<td>82</td>
</tr>
</tbody>
</table>

\[ \prod_{i \in V} u_i(A_i) = (55 + 45 + 15) \cdot (30 + 22 + 17) \cdot (82 + 58) \]

\[ \prod_{i \in V} u_i(B_i) = (45 + 15 + 35 + 50) \cdot (17 + 90) \cdot (82 + 58) \]

\[ \prod_{i \in V} u_i(B_i) > \prod_{i \in V} u_i(A_i) \]
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i).
\]

(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)

Proposition: MNW satisfies PO and EF1.
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:
\[
\prod_{i \in V} u_i(A_i). \quad (\text{If in each allocation the utility of some voters is zero, do some clever tie-breaking.})
\]

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{ A_i \}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \text{ (If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).

PO is straightforward. Why?
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i).
\]

(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i).
\]

(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).

For EF1: assume that an allocation \( \{A_i\}_{i \in V} \) that maximises \( \prod_{i \in V} u_i(A_i) \) is not EF1.
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:
\[
\prod_{i \in V} u_i(A_i). \text{ (If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).
For EF1: assume that an allocation \( \{A_i\}_{i \in V} \) that maximises \( \prod_{i \in V} u_i(A_i) \) is not EF1.

Let \( v_i \) and \( v_j \) be such that
\[
u_i(A_j) > u_i(A_i) + u_i(o) \text{ for each } o \in A_j. \quad (1)
\]
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \quad \text{(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).

For EF1: assume that an allocation \( \{A_i\}_{i \in V} \) that maximises \( \prod_{i \in V} u_i(A_i) \) is not EF1.

Let \( v_i \) and \( v_j \) be such that \( u_i(A_j) > u_i(A_i) + u_i(o) \) for each \( o \in A_j \).

Let \( g \) be such an object from \( A_j \) that \( u_j(g)/u_i(g) \leq u_j(A_j)/u_i(A_j) \).
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:
\[
\prod_{i \in V} u_i(A_i). \quad \text{(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).

For EF1: assume that an allocation \( \{A_i\}_{i \in V} \) that maximises \( \prod_{i \in V} u_i(A_i) \) is not EF1.

Let \( v_i \) and \( v_j \) be such that \( u_i(A_j) > u_i(A_i) + u_i(o) \) for each \( o \in A_j \). \hspace{1cm} (1)

Let \( g \) be such an object from \( A_j \) that \( u_j(g)/u_i(g) \leq u_j(A_j)/u_i(A_j) \). \hspace{1cm} (2)

(say the one with the minimal ratio \( u_j(g)/u_i(g) \))
Is there an allocation that is EF1 and PO?

**Maximise Nash Welfare (MNW):** select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \quad (\text{If in each allocation the utility of some voters is zero, do some clever tie-breaking.})
\]

**Proposition:** MNW satisfies PO and EF1.

**Proof:** (the case of non-zero utilities in an allocation returned by MNW).

For EF1: assume that an allocation \( \{A_i\}_{i \in V} \) that maximises \( \prod_{i \in V} u_i(A_i) \) is not EF1.

Let \( v_i \) and \( v_j \) be such that \( u_i(A_j) > u_i(A_i) + u_i(o) \) for each \( o \in A_j \). \hspace{1cm} (1)

Let \( g \) be such an object from \( A_j \) that \( u_j(g)/u_i(g) \leq u_j(A_j)/u_i(A_j) \). \hspace{1cm} (2)

(say the one with the minimal ratio \( u_j(g)/u_i(g) \))

After multiplying (1) and (2):

\[
(u_j(g)/u_i(g)) \cdot (u_i(A_i) + u_i(g)) < (u_j(A_j)/u_i(A_j)) \cdot u_i(A_j).
\]
Is there an allocation that is EF1 and PO?

**Maximise Nash Welfare (MNW):** select an allocation \(\{A_i\}_{i \in V}\) that maximises:
\[
\prod_{i \in V} u_i(A_i). \quad \text{(If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

**Proposition:** MNW satisfies PO and EF1.

**Proof:** (the case of non-zero utilities in an allocation returned by MNW).

For EF1: assume that an allocation \(\{A_i\}_{i \in V}\) that maximises \(\prod_{i \in V} u_i(A_i)\) is not EF1.

Let \(v_i\) and \(v_j\) be such that \(u_i(A_j) > u_i(A_i) + u_i(o)\) for each \(o \in A_j\). \hspace{1cm} (1)

Let \(g\) be such an object from \(A_j\) that \(u_j(g)/u_i(g) \leq u_j(A_j)/u_i(A_j)\). \hspace{1cm} (2)

(say the one with the minimal ratio \(u_j(g)/u_i(g)\))

After multiplying (1) and (2):

\[
(u_j(g)/u_i(g)) \cdot (u_i(A_i) + u_i(g)) < (u_j(A_j)/u_i(A_j)) \cdot u_i(A_j).
\]

This is equivalent to: \(u_j(A_j) \cdot u_i(g) > u_i(g) \cdot u_j(g) + u_i(A_i) \cdot u_j(g)\).
Is there an allocation that is EF1 and PO?

Maximise Nash Welfare (MNW): select an allocation \( \{A_i\}_{i \in V} \) that maximises:

\[
\prod_{i \in V} u_i(A_i). \text{ (If in each allocation the utility of some voters is zero, do some clever tie-breaking.)}
\]

Proposition: MNW satisfies PO and EF1.

Proof: (the case of non-zero utilities in an allocation returned by MNW).

For EF1: assume that an allocation \( \{A_i\}_{i \in V} \) that maximises \( \prod_{i \in V} u_i(A_i) \) is not EF1.

Let \( v_i \) and \( v_j \) be such that \( u_i(A_j) > u_i(A_i) + u_i(o) \) for each \( o \in A_j \). \hspace{1cm} (1)

Let \( g \) be such an object from \( A_j \) that \( u_j(g)/u_i(g) \leq u_j(A_j)/u_i(A_j) \). \hspace{1cm} (2)

(say the one with the minimal ratio \( u_j(g)/u_i(g) \))

After multiplying (1) and (2):

\[
(u_j(g)/u_i(g)) \cdot (u_i(A_i) + u_i(g)) < (u_j(A_j)/u_i(A_j)) \cdot u_i(A_j).
\]

This is equivalent to: \( u_j(A_j) \cdot u_i(g) > u_i(g) \cdot u_j(g) + u_i(A_i) \cdot u_j(g) \).

This, on the other hand is equivalent to:

\[
(u_i(A_i) + u_i(g)) \cdot (u_j(A_j) - u_j(g)) > u_i(A_i) \cdot u_j(A_j).
\]
What if the valuations are not additive?
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each $v_i \in V$ and all $S' \subseteq S \subseteq \emptyset$ we have $u_i(S') \leq u_i(S)$. 
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \emptyset \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \emptyset \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object \( o \in \emptyset \) at hand do the following:

1. Ensure that there is a voter \( v \in V \) that no other voter envies.
2. Assign \( o \in \emptyset \) to \( v \).
What if the valuations are not additive?

**Only** assume that the utility function of each voter is *monotonic*:
for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \emptyset \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order.
Having an object \( o \in \emptyset \) at hand do the following:
1. Ensure that there is a voter \( v \in V \) that no other voter envies.
2. Assign \( o \in \emptyset \) to \( v \).

**How to ensure there is an unenvied voter?**
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each $v_i \in V$ and all $S' \subseteq S \subseteq \emptyset$ we have $u_i(S') \leq u_i(S)$.

Can we now achieve EF1?

**Envy-graph procedure (EGP):** Consider objects in some predefined order. Having an object $o \in \emptyset$ at hand do the following:

1. Ensure that there is a voter $v \in V$ that no other voter envies.
2. Assign $o \in \emptyset$ to $v$.

How to ensure there is an unenvied voter?

Assume there is no such voter.
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic:
for each $v_i \in V$ and all $S' \subseteq S \subseteq \emptyset$ we have $u_i(S') \leq u_i(S)$.

Can we now achieve EF1?

Envy-graph procedure (EGP): consider objects in some predefined order.
Having an object $o \in \emptyset$ at hand do the following:

1. Ensure that there is a voter $v \in V$ that no other voter envies.
2. Assign $o \in \emptyset$ to $v$.

How to ensure there is an unenvied voter?

Assume there is no such voter. Then, there is a cycle of envy: $v_{i_1}, v_{i_2}, \ldots, v_{i_p}$ such that $v_{i_1}$ envies $v_{i_2}$, $v_{i_2}$ envies $v_{i_3}$, ..., $v_{i_{p-1}}$ envies $v_{i_p}$, and $v_{i_p}$ envies $v_{i_1}$. 
What if the valuations are not additive?

**Only** assume that the utility function of each voter is **monotonic**:
for each $v_i \in V$ and all $S' \subseteq S \subseteq \emptyset$ we have $u_i(S') \leq u_i(S)$.

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object $o \in \emptyset$ at hand do the following:
1. Ensure that there is a voter $v \in V$ that no other voter envies.
2. Assign $o \in \emptyset$ to $v$.

**How to ensure there is an unenvied voter?**

Assume there is no such voter. Then, there is a cycle of envy: $v_{i_1}, v_{i_2}, \ldots, v_{i_p}$ such that $v_{i_1}$ envies $v_{i_2}$, $v_{i_2}$ envies $v_{i_3}$, $v_{i_3}$ envies $v_{i_4}$, ..., $v_{i_{p-1}}$ envies $v_{i_p}$, and $v_{i_p}$ envies $v_{i_1}$.

We remove this cycle by: giving $v_{i_1}$ the bundle of $v_{i_2}$, giving $v_{i_2}$ the bundle of $v_{i_3}$, ..., giving $v_{i_p}$ the bundle of $v_{i_1}$. 
What if the valuations are not additive?

**Only** assume that the utility function of each voter is **monotonic**: for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \mathcal{O} \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object \( o \in \mathcal{O} \) at hand do the following:

1. Ensure that there is a voter \( v \in V \) that no other voter envies.
2. Assign \( o \in \mathcal{O} \) to \( v \).

How to ensure there is an unenvied voter?

Assume there is no such voter. Then, there is a cycle of envy: \( v_{i_1}, v_{i_2}, \ldots v_{i_p} \) such that \( v_{i_1} \) envies \( v_{i_2} \), \( v_{i_2} \) envies \( v_{i_3} \), \ldots, \( v_{i_{p-1}} \) envies \( v_{i_p} \), and \( v_{i_p} \) envies \( v_{i_1} \).

We remove this cycle by: giving \( v_{i_1} \) the bundle of \( v_{i_2} \), giving \( v_{i_2} \) the bundle of \( v_{i_3} \), \ldots, giving \( v_{i_p} \) the bundle of \( v_{i_1} \). If there is still a cycle, we repeat the procedure, etc.
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \mathcal{O} \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object \( o \in \mathcal{O} \) at hand do the following:
1. Ensure that there is a voter \( v \in V \) that no other voter envies.
2. Assign \( o \in \mathcal{O} \) to \( v \).

How to ensure there is an unenvied voter?

Assume there is no such voter. Then, there is a cycle of envy: \( v_{i_1}, v_{i_2}, \ldots, v_{i_p} \) such that \( v_{i_1} \) envies \( v_{i_2} \), \( v_{i_2} \) envies \( v_{i_3} \), \ldots, \( v_{i_{p-1}} \) envies \( v_{i_p} \), and \( v_{i_p} \) envies \( v_{i_1} \).

We remove this cycle by: giving \( v_{i_1} \) the bundle of \( v_{i_2} \), giving \( v_{i_2} \) the bundle of \( v_{i_3} \), \ldots, giving \( v_{i_p} \) the bundle of \( v_{i_1} \). If there is still a cycle, we repeat the procedure, etc. Why does this procedure finish?
What if the valuations are not additive?

Only assume that the utility function of each voter is **monotonic**: for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \emptyset \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object \( o \in \emptyset \) at hand do the following:

1. Ensure that there is a voter \( v \in V \) that no other voter envies.
2. Assign \( o \in \emptyset \) to \( v \).

Why this procedure is EF1?
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each $v_i \in V$ and all $S' \subseteq S \subseteq \emptyset$ we have $u_i(S') \leq u_i(S)$.

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object $o \in \emptyset$ at hand do the following:
1. Ensure that there is a voter $v \in V$ that no other voter envies.
2. Assign $o \in \emptyset$ to $v$.

**Why this procedure is EF1?**

After each step it produces an EF1 allocation:
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \varnothing \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object \( o \in \varnothing \) at hand do the following:
1. Ensure that there is a voter \( v \in V \) that no other voter envies.
2. Assign \( o \in \varnothing \) to \( v \).

**Why this procedure is EF1?**

After each step it produces an EF1 allocation:
1. Removing cycles does not change the fact that the allocation is envy free:
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic: for each \( v_i \in V \) and all \( S' \subseteq S \subseteq \emptyset \) we have \( u_i(S') \leq u_i(S) \).

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order. Having an object \( o \in \emptyset \) at hand do the following:
1. Ensure that there is a voter \( v \in V \) that no other voter envies.
2. Assign \( o \in \emptyset \) to \( v \).

Why this procedure is EF1?

After each step it produces an EF1 allocation:
1. Removing cycles does not change the fact that the allocation is envy free:
   - The bundles are the same, they are only redistributed, and
   - After removing a cycle each voter gets at least as good bugle as before.
What if the valuations are not additive?

Only assume that the utility function of each voter is monotonic:
for each $v_i \in V$ and all $S' \subseteq S \subseteq \emptyset$ we have $u_i(S') \leq u_i(S)$.

Can we now achieve EF1?

**Envy-graph procedure (EGP):** consider objects in some predefined order.

Having an object $o \in \mathcal{O}$ at hand do the following:
1. Ensure that there is a voter $v \in V$ that no other voter envies.
2. Assign $o \in \mathcal{O}$ to $v$.

Why this procedure is EF1?

After each step it produces an EF1 allocation:
1. Removing cycles does not change the fact that the allocation is envy free:
   - The bundles are the same, they are only redistributed, and
   - After removing a cycle each voter gets at least as good bugle as before.
2. We give an item to a voter that is unenvied. The only envy introduced could be towards this voter. Yet if we remove the item the voter was given, then the envy towards her disappears.
Further Topics

Allocation of bads. What if the utilities are negative?
Further Topics

**Allocation of bads.** What if the utilities are negative?
1. Then the EF1 allocation always exists. If you are interested, see:

Further Topics

Allocation of bads. What if the utilities are negative?
1. Then the EF1 allocation always exists. If you are interested, see:


2. Open problem: does there exist an EF1 and PO allocation?
Further Topics

Allocation of bads. What if the utilities are negative?
1. Then the EF1 allocation always exists. If you are interested, see:


2. Open problem: does there exist an EF1 and PO allocation?

Stronger concepts of fairness:
Further Topics

Allocation of bads. What if the utilities are negative?
1. Then the EF1 allocation always exists. If you are interested, see:


2. Open problem: does there exist an EF1 and PO allocation?

Stronger concepts of fairness:

Envy-freeness up to any good (EFX): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free up any good if for each \( v_i, v_j \in V \) and each \( o \in A_j \) we have
\[
    u_i(A_i) \geq u_i(A_j \setminus \{o\}).
\]
Further Topics

**Allocation of bads.** What if the utilities are negative?

1. Then the EF1 allocation always exists. If you are interested, see:


2. **Open problem:** does there exist an EF1 and PO allocation?

Stronger concepts of fairness:

**Envy-freeness up to any good (EFX):** We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free up any good if for each \( v_i, v_j \in V \) and each \( o \in A_j \) we have

\[
u_i(A_i) \geq u_i(A_j \setminus \{o\}).\]

**Open problem:** does there always exist an EFX allocation?
Further Topics

Allocation of bads. What if the utilities are negative?
1. Then the EF1 allocation always exists. If you are interested, see:


2. Open problem: does there exist an EF1 and PO allocation?

Stronger concepts of fairness:

Envy-freeness up to any good (EFX): We say that an allocation \( \{A_i\}_{i \in V} \) is envy-free up any good if for each \( v_i, v_j \in V \) and each \( o \in A_j \) we have
\[
u_i(A_i) \geq u_i(A_j \setminus \{o\}).
\]

Open problem: does there always exist an EFX allocation?

We will also discuss fairness concepts for divisible goods such as competitive equilibria, which apply to indivisible goods.