Algorithmic Game Theory - Fair Division	*Martin Hoefer, Giovanna Varricchio*

Fair Division of Indivisible Goods

Summary: We introduce fundamental ideas of fair division for the indivisible setting. Here multiple items have to be assigned. Unlike in cake cutting, items are indivisible, i.e., each item can be assigned to at most one agent. We consider relaxations of classical fairness notions and explore their computation. We also consider efficiency in conjunction with fairness.

Resources:

- *Fair Division of Indivisible Goods: A Survey* G. Amanatidis, G. Birmpas, A. Filos-Ratsikas, A. A. Voudouris `https://arxiv.org/pdf/2202.07551.pdf`
- *Trends in computational social choice.* Chapter 12 – Approximation Algorithms and Hardness Results for Fair Division with Indivisible Goods.
- *Tutorial on Recent Advances in Fair Resource Allocation*, Rupert Freeman and Nisarg Shah `https://www.cs.toronto.edu/~nisarg/papers/Fair-Division-Tutorial.pdf`
- Further readings in the references.

1 Setting

We are given a set of m indivisible resources, a.k.a. *items* or *goods*, $\mathcal{G} = \{g_1, \dots, g_m\}$, and a set $\mathcal{N} = \{1, \dots, n\}$ of n agents.

Definition 1 (Allocation)**.** *An* allocation $\mathcal{A}$ *is a partition of* $\mathcal{G}$ *into disjoint sets, each of them assigned to at most one agent. For each* $i \in \mathcal{N}$*, we denote by* $A_i \subseteq \mathcal{G}$ *the* bundle *(that is, the set of items) received by agent* i *in the allocation* $\mathcal{A}$*.*

Usually, the allocation is also asked to be complete, that is, $\cup_i A_i = \mathcal{G}$.

Agents' valuations. Agents have preferences over possible bundles they might receive. Preferences are usually quantifiable and are expressed by means of valuations functions.

Definition 2 (Valuations)**.** *The* valuation function *of agent* i *is a mapping* $v_i : 2^{\mathcal{G}} \to \mathbb{R}_{\geq 0}$.

We will mostly focus on valuation functions that are additive. The results we provide hold only for additive valuations unless specified otherwise.

Definition 3 (Additive Valuations)**.** *A valuation function* $v : 2^{\mathcal{G}} \to \mathbb{R}_{\geq 0}$ *is called* additive *if for each* $X \subseteq \mathcal{G}$, $v(X) = \sum_{g \in X} v(\{g\})$.

For our convenience, in what follows we will write $v(g)$ instead of $v(\{g\})$.

Example 1. *Consider the example of an instance with additive valuations depicted in Table 1. We have three agents and five items. The depicted allocation gives a value of* 17 *to agent 1, 12 to agent 2, and 3 to agent 3.*

2 Fairness Criteria

In this section, we reintroduce some fairness criteria from cake cutting. We will see that the solutions we defined are no longer guaranteed to exist. Therefore, we introduce some relaxations to circumvent this problem.

	g_1	g_2	g_3	g_4	g_5
Agent 1	15	3	2	2	6
Agent 2	7	5	5	5	7
Agent 3	20	3	3	3	3

Table 1: An example of additive valuations. Line i corresponds to agent i. Circles correspond to the allocated items in a possible (complete) allocation.

2.1 Definitions

The *proportional share* of agent i is given by $\mathsf{PS}_i = \frac{v_i(\mathcal{G})}{n}$.

Definition 4 (Proportionality)**.** *An allocation $\mathcal{A}$ is called* proportional (PROP) *if each agent receives at least her proportional share, that is, $\forall i \in \mathcal{N}$ it holds*

$$v_i(A_i) \geq \mathsf{PS}_i \ .$$

Also redefining envy-freeness is straightforward.

Definition 5 (Envy-freeness)**.** *An allocation $\mathcal{A}$ is called* envy-free (EF) *if for each $i, j \in \mathcal{N}$ it holds*

$$v_i(A_i) \geq v_i(A_j) \ .$$

Observe that for indivisible items EF $\implies$ PROP, and there might exist allocations which are PROP but not EF.
Unfortunately, EF and PROP allocations may not exist.

Example 2. *Let us consider two agents and one valuable (positively valued by both agents) item. No matter who receives the item the resulting allocation is neither* EF *nor* PROP*.*
Such an impossibility holds even when $m > n$, see our example in Table 1. Indeed, this instance does not admit any envy-free or proportional allocation. Consider agent 3. She must get at least $\{g_1\}$ or $\{g_2, g_3, g_4, g_5\}$. The latter case, leaves one item to be allocated to either agent 1 or 2 which cannot lead to a proportional allocation. On the other hand, if agent 3 gets g_1, agent 1 must receive at least three of the remaining four goods and 2 must get at least two, which is not possible.

Due to this impossibility result, relaxed versions of EF and PROP have been introduced and studied.

Definition 6 (Proportionality up to one Good)**.** *An allocation $\mathcal{A}$ is called* proportional up to one good (PROP1) *if, for each $i \in \mathcal{N}$, either $A_i = \mathcal{G}$ or there exists $g \in \mathcal{G} \setminus A_i$*

$$v_i(A_i \cup \{g\}) \geq \mathsf{PS}_i \ .$$

Notice that $v_i(A_i \cup g) = v_i(A_i) + v_i(g)$ when valuations are additive.

Definition 7 (Envy-Freeness up to one Good)**.** *An allocation $\mathcal{A}$ is called* envy-free up to one good (EF1) *if, for each $i, j \in \mathcal{N}$, either $A_j = \emptyset$ or there exists $g \in A_j$ such that*

$$v_i(A_i) \geq v_i(A_j \setminus \{g\}) \ .$$

Example 3. *The allocation depicted in Table 1 is an* EF1 *allocation.*

Remark Clearly, $\mathsf{EF} \implies \mathsf{EF1}$ and $\mathsf{PROP} \implies \mathsf{PROP1}$.

Proposition 1. *Any* EF1 *allocation is also* PROP1.

Proof. Let us show the statement for agent i. Since i is EF1, for each $j \in \mathcal{N}$, with $A_j \neq \emptyset$, there exists $g_j \in A_j$ such that

$$v_i(A_i) \geq v_i(A_j \setminus \{g_j\}) \ .$$

Summing up for all $i \in \mathcal{N}$, and by additivity, we get

$$n \cdot v_i(A_i) \geq v_i(A_i) + \sum_{j \neq i} (v_i(A_j) - v_i(g_j)) = v_i(\mathcal{G}) - \sum_{j \neq i} v_i(g_j) \ ,$$

and therefore $v_i(A_i) \geq \mathsf{PS}_i - \frac{\sum_{j \neq i} v_i(g_j)}{n}$. If every $A_j = \emptyset$, then $A_i = \mathcal{G}$ making the allocation proportional for i. Otherwise, by selecting $g^* = \arg\max_{g \in \mathcal{G} \setminus A_i} v_i(g)$, we have $v_i(A_i) \geq \mathsf{PS}_i - \frac{(n-1)v_i(g^*)}{n} \geq \mathsf{PS}_i - v_i(g^*)$, and hence agent i is PROP1.
Applying the same argument to all the agents the statement follows. □

Proposition 2. *There exist allocations that are* PROP1 *but not* EF1.

Proof. Exercise. □

3 Existence and Computation of EF1 and PROP1 Allocations

let us discuss the computation of EF1 (and hence PROP1) allocations for additive and monotone valuation functions.

3.1 Round-Robin Procedure

We show that, thanks to a Round-Robin procedure, EF1 allocations always exist for additive valuations and can be computed in polynomial time.
Let us first consider general sequential algorithms. Roughly speaking, in a sequential allocation of items, we create a vector (sequence) $s = (s_1, \ldots, s_m)$ where the component s_h corresponds to the agent who will select her most preferred item at the h-th round of the procedure. The vector s is also known as *picking sequence*.

Sequential allocation algorithm. A sequential allocation algorithm takes as input a picking sequence s, the goods $\mathcal{G}$, the agents $\mathcal{N}$, and their valuations. The algorithm proceeds as follows:

- $\mathcal{A} \leftarrow (\emptyset, \ldots, \emptyset)$
- For $h = 1, \ldots, m$
  - $i \leftarrow s_h$
  - $g^* \leftarrow \arg\max_{g \in \mathcal{G}} v_i(g)$
  - $A_i \leftarrow A_i \cup \{g^*\}$, $\mathcal{G} \leftarrow \mathcal{G} \setminus \{g^*\}$

The *Round-Robin procedure* is the sequential allocation algorithm executed with a picking sequence of length m in the form $s = (1, \ldots, n, 1, \ldots, n, 1, \ldots)$, for a fixed ordering $1, \ldots, n$ of the agents.

Example 4. *The allocation in Tab. 1 is the result of Round-Robin for the ordering* $1, 2, 3$.

Theorem 3. *The Round-Robin procedure outputs an* EF1 *allocation when agents' valuations are additive.*

Proof. Let us split the algorithm into rounds: We call Round k the k-th occurrence of $1, \dots, n$ in the picking sequence $s = (1, \dots, n, 1, \dots, n, 1, \dots)$. Therefore, in a round k, the agents receive a k-th item, if possible. Notice that in the last round it is possible that not all the agents have the opportunity to select an item.
We start by noticing that the first agent in the ordering (which is 1) is EF since the item she gets in a Round k is at least as good as the item selected by any other agent in the same round.
Let us consider agent i and remove the first $i-1$ agents in the sequence s (let's call this new sequence $s(i)$) and remove the items these agents selected in the first round. By running Round-Robin with $s(i)$ on the refined set of goods i is the first agent in the sequence. Notice that $s(i)$ is a Round-Robin sequence for the ordering $i, i+1, \dots n, 1, \dots i-1$, and hence we get an EF allocation for i. By reassigning the items we removed to their owner we get an $\mathsf{EF1}$ allocation. Moreover, this allocation coincides with the outcome of the original Round-Robin with the sequence s, and therefore the statement follows. □

3.2 Envy-Graph and Envy-Cycle Elimination

Is it possible to achieve $\mathsf{EF1}$ for more general valuation functions?

Definition 8 (Monotone Valuations)**.** *A valuation function $v : 2^{\mathcal{G}} \to \mathbb{R}_{\geq 0}$ is called* monotone *if for each $Y \subseteq X \subseteq \mathcal{G}$, $v(Y) \leq v(X)$.*

For our purposes we need to introduce the following instrument:

Definition 9 (Envy-Graph)**.** *Given a (partial) allocation $\mathcal{A}$, the* envy graph *for the allocation $\mathcal{A}$ is defined as follows:*

- *each agent i is represented by a node, for simplicity we call the node i;*
- *there exists a directed edge (i, j) if and only if $v_i(A_j) > v_i(A_i)$.*

Note that the directed edge represents the envy of i towards j. Hence, if i is a source in the envy-graph, she is not envied by any other agent.
The envy-graph is an extremely useful tool to reduce envy in an allocation. In fact, if there exists a cycle we can reduce the envy by trading bundles along the cycle. Formally, let us assume that an allocation $\mathcal{A}$ induces a cycle $C = i_1, i_2, \dots, i_k, i_1$ involving k (w.l.o.g.) distinct agents. Trading the bundles along the cycle means we create a new allocation $\mathcal{A}'$ where $A'_i = A_{i+1}$ for each $i = 1, \dots k-1$ and $A'_k = A_1$ while all the other bundles remains the same. By trading bundles along a cycle we reduce the number of edges in the envy-graph without creating new ones. More importantly, trading along an envy-cycle preserves the $\mathsf{EF1}$ property, as formalized in the following lemma:

Lemma 4. *Given an $\mathsf{EF1}$ allocation $\mathcal{A}$, if the envy-graph has a cycle C, then the allocation $\mathcal{A}'$ obtained from $\mathcal{A}$ by trading the cycle C is also $\mathsf{EF1}$.*

Proof. From the perspective of agents who are not in the cycle, the allocation is not changing significantly (we are only changing the owners of the bundles).
For the agents in the cycle, the value of their bundle increases. It is higher than the value attributed to the bundle they previously had. Therefore, in the new allocation, any agent i in the cycle does not envy her previous bundle while the previous edges (i, j) can only disappear. □

The envy-cycle elimination protocol. The envy-cycle elimination starts from an empty allocation (which is clearly $\mathsf{EF1}$). At each round, one available item g is allocated to some agent i who is not envied by any other (which is a source node in the envy-graph). This maintains the $\mathsf{EF1}$ property since i was note envied before inserting g. At the end of the round, to guarantee the existence of source nodes, if a cycle appears in the envy-graph, bundles along the cycle are traded. Therefore, at the end of each round there is no cycle in the envy-graph. Since $\mathsf{EF1}$ is preserved (by Lemma 4) at each step of the procedure, the final allocation is $\mathsf{EF1}$.
Formally, envy-cycle elimination can be summarized as follows:

- $\mathcal{A} \leftarrow (\emptyset, \dots, \emptyset)$
- Sort goods from g_1 up to g_m
- For $h = 1, \dots, m$
  - $i \leftarrow$ a sink node in the envy-graph
  - $A_i \leftarrow A_i \cup \{g_h\}$, $\mathcal{G} \leftarrow \mathcal{G} \setminus \{g_h\}$
  - update the envy-graph
  - While there exists cycle C in the envy-graph:
    - trade along cycle C

Theorem 5 (Lipton et al. 2004)**.** *If agents' valuations are monotone, then envy-cycle elimination outputs an* EF1 *allocation.*

Proof. The proof proceeds by induction as explained above. At each step the EF1 property is satisfied. □

What is the complexity of envy-cycle elimination? This procedure guarantees the existence of an EF1 allocation for monotone valuations; however, the time complexity of the procedure is not clear. This is closely related to the representation and the knowledge we have of the valuation functions.
At every iteration of the FOR-loop, after adding in the next item, the envy graph has to be updated, and therefore all agents' valuation functions have to be evaluated on the new bundle. The evaluation cannot be considered an atomic operation unless we have an oracle. Let us denote by T^* the time complexity for determining the value of any bundle by any agent. Then the runtime caused by the parts of the FOR loop not including the WHILE loop is $O(mnT^*)$.
To bound the overall number of cyclic trades within the WHILE loop, note that by performing a cycle, at least two edges disappear in the envy-graph. How many edges are introduced during the whole execution? In every iteration of the FOR loop at most $n-1$ new edges are introduced. Hence overall at most $O(mn)$ edges are introduced, and thus the number of cycles performed is also at most $O(mn)$. Performing a cycle can be done in $O(n^2)$, e.g., by using adjacency matrices for graph representation. The complexity of envy-cycle elimination is thus $O(mnT^* + n^3m)$, and this can be considered polynomial-time, since T^* is an intrinsic value depending on the given valuations.

4 EF1 and Efficiency

We now turn our attention again to additive valuations. Since EF1 allocations do always exist, we may try to ask for further properties for such a solution. A compelling notion is the one of efficiency, usually defined as Pareto optimality. Roughly speaking, we do not want to create waste while achieving EF1.
We have seen that maximizing welfare functions such as the utilitarian or the Nash social welfare leads to Pareto optimal allocations in cake cutting. It remains true also for indivisible goods (it is sufficient to apply the very same arguments). Interestingly, it turns out that any allocation maximizing Nash social welfare is also particularly fair.

Remark. There are scenarios in which the maximum Nash social welfare is 0. Consider for example the case of two agents and an item. In this case, we at first maximize the number of agents having positive value for their bundle, then maximize the Nash welfare among these agents. We call such allocations *Nash optimal.*

Theorem 6. *Let $\mathcal{A}$ be a Nash optimal allocation, then $\mathcal{A}$ is also* EF1*.*

Proof. Let be $\mathcal{A}$ a Nash optimal allocation.
Let us assume $\text{NSW}(\mathcal{A}) \neq 0$, which means, every agent has a positive value for her bundle, and therefore no bundle is empty. It is possible to show that the statement holds true even if $\text{NSW}(\mathcal{A}) = 0$ with a careful adaptation of the proof.
We want to show that for every $i, j \in \mathcal{N}$, there exists $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j) - v_i(g)$.
If $v_i(A_j) = 0$ the claim trivially follows since i does not envy j.
Otherwise, since $\mathcal{A}$ is Nash optimal, by moving any item $g \in A_j$ to A_i the Nash social welfare cannot improve. Hence,

$$\begin{aligned} v_i(A_i) \cdot v_j(A_j) \geq (v_i(A_i) + v_i(g)) \cdot (v_j(A_j) - v_j(g)) \qquad & \Leftrightarrow \\ v_j(g) \cdot (v_i(A_i) + v_i(g)) \geq v_i(g) \cdot v_j(A_j) \qquad & \end{aligned}$$

showing that, for each $g \in A_j$

$$v_i(A_i) + v_i(g) \geq \frac{v_i(g)}{v_j(g)} \cdot v_j(A_j) \ . \tag{1}$$

Let us select $g^* = \arg\min_{g \in A_j, v_i(g) > 0} \frac{v_j(g)}{v_i(g)}$. Notice that g^* is well defined; in fact, there must exist at least one positively valued good in j's bundle according to i's valuations because $v_i(A_j) > 0$. By definition of g^*, it holds

$$\frac{v_j(g^*)}{v_i(g^*)} \leq \frac{\sum_{g \in A_j} v_j(g)}{\sum_{g \in A_j} v_i(g)} \leq \frac{v_j(A_j)}{v_i(A_j)}$$

and hence, by inverting terms, $\frac{v_i(A_j)}{v_j(A_j)} \leq \frac{v_i(g^*)}{v_j(g^*)}$.
This inequality together with (1)) shows that $g^* \in A_j$ is such that $v_i(A_i) \geq v_i(A_j) - v_i(g^*)$, concluding the proof. □

In conclusion, for additive valuations there exists an allocation that is simultaneously EF1 and Pareto optimal.

Proposition 7. *Under additive valuations, an allocation that is simultaneously* EF1 *and PO always exists.*

This is only an existential result, computing a maximum NSW allocation is in general hard, even for two agents with identical valuations.
So far we discussed the existence of EF1 and hence PROP1 allocations. Are there any other meaningful relaxations for EF and PROP?

5 Beyond EF1 Allocations – Envy-Freeness up to any Good

EF1 represents in first approximation a good relaxation of the EF fairness concept. While we know that EF $\implies$ EF1, we see an example in which an EF1 allocation might be quite unfair.

Example 5. *Consider a fair division instance with two agents, three goods, and additive valuations depicted in Table 2.*
Let us consider the allocation $A_1 = \{g_1, g_3\}$ and $A_2 = \{g_2\}$. This allocation is EF1 *as agent 2 is allowed to remove g_1 to eliminate the envy. However, according to agent 2, $v_2(A_1) = 13$ and $v_2(A_2) = 3$ which is a quite large gap between the two bundles.*

Roughly speaking, in the definition of EF1, allowing to remove *some* good could be too "generous", what about the removal of *any*?

Definition 10 (Envy-freeness up to Any Good)**.** *An allocation $\mathcal{A}$ is called* envy-free up to any good (EFX) *if, for each $i, j \in \mathcal{N}$, either $A_j = \emptyset$ or for every $g \in A_j$ such that $v_i(g) > 0$ it holds*

$$v_i(A_i) \geq v_i(A_j \setminus \{g\}) \ .$$

	g_1	g_2	g_3
Agent 1	10	2	1
Agent 2	10	3	2

Table 2: Valuations in Example 5.

Example 6. *Consider the allocation in Table 1. The allocation is* EF1 *but not* EFX*.*
Consider for the same instance the allocation $A_1 = \{g_4, g_5\}$*,* $A_2 = \{g_2, g_3\}$*,* $A_3 = \{g_1\}$*. This is an* EFX *allocation.*

The following implications easily follow.

Proposition 8. EF $\implies$ EFX $\implies$ EF1

It is also easy to see that backward directions do not hold.

5.1 On EFX Existence

Unfortunately, the existence of EFX-allocations is unknown, even for additive valuations! It is only known to be guaranteed for special cases like two or three agents or identical valuations.

Two agents. We show the existence of EFX allocations for two agents.

Theorem 9. EFX *allocations always exists for* $n = 2$ *and can be efficiently computed.*

Proof. We show the existence by providing an algorithm that is a discrete version of the CutAndChoose protocol.

- Agent 1 computes a partition (A_1, A_2) such that $v_1(A_1) \geq v_1(A_2)$ and $v_1(A_1 \setminus \{g\}) \leq v_1(A_2)$ for each $g \in A_1$ such that $v_1(g) > 0$;
- Agent 2 selects the most favorite bundle for her;
- Agent 1 gets the remaining bundle.

The resulting allocation is EFX. Indeed, agent 2 does not envy agent 1. Agent 1, no matter which bundle she receives, is EFX by the conditions on the two bundles. We only need to clarify how to compute such a partition.
To compute the partition (A_1, A_2) we proceed as follows[1]:

- Sort goods in a non-increasing order of values according to 1, that is, $v_1(g_1) \geq v_1(g_2) \geq \ldots \geq v_1(g_m)$
- $(A_1, A_2) \leftarrow (\emptyset, \emptyset)$
- allocate items in the ordering $g_1 \ldots g_m$ to the bundle A_i, $i = 1, 2$, of minimum value for agent 1.

Assume without loss of generality that $v_1(A_1) \geq v_1(A_2)$, otherwise we change names to the bundles. We have that $v_1(A_1 \setminus \{g\}) \leq v_1(A_2)$ for each $g \in A_1$, such that $v_1(g) > 0$, must hold. Notice that this means that for 1, receiving A_2 would make her EFX. Hence, it is sufficient to show that $v_1(A_1 \setminus \{g^*\}) \leq v_1(A_1)$ for $g^* \in A_1$ such that $v_1(g^*) > 0$ and g^* is the least valued good in A_1. This holds true since before inserting g^* in A_1 the value of A_1 was smaller or equal to the value of A_2, according to 1. Furthermore, g^* is the smallest positively valued good in A_1, therefore the statement follows. ☐

[1] Notice we will use the same idea in the next paragraph for identical valuations.

Identical additive valuations. Now we assume that the agents have identical additive valuations, that is, $v_i = v$ for each $i \in \mathcal{N}$ where v is additive.
We start by showing that EFX allocations always exist in this setting.

Theorem 10. *If agents have identical valuations, every Nash optimal allocation is* EFX.

Proof. Assume $\mathcal{A}$ is Nash optimal. We denote by v the valuation function of the agents. Therefore, moving any positively valued g from any A_j to any A_i cannot strictly improve the Nash social welfare. Formally,

$$\begin{aligned} v(A_i) \cdot v(A_j) \geq (v(A_i) + v(g)) \cdot (v(A_j) - v(g)) \qquad & \Leftrightarrow \\ v(g) \cdot (v(A_i) + v(g)) \geq v(g) \cdot v_j(A_j) \,. \end{aligned}$$

Since $v(g) > 0$, we have for each i, j and $g \in A_j$, $v(A_i) \geq v_j(A_j) - v(g)$ and the statement follows.
Notice we assumed that the NSW is not 0. If so, it means that at least one agent has no item in her bundle, and hence there are not enough items for the agents. Recall that in this case, we assume that we first maximize the number of agents with a positive value for their bundle and then the Nash welfare among those agents. Such an allocation assigns each agent at most one item, and hence it is EFX. ☐

Unfortunately, it is hard to compute such an allocation, even for $n = 2$.

Theorem 11. *It is NP-hard to compute a Nash optimal allocation, even with identical valuations and* $n = 2$.

Proof. Reduction from PARTITION.

PARTITION
Input: A set $X = \{x_1, \dots, x_t\}$ of positive values **Problem:** Does there exist a partition of X, i.e. $(S, X \setminus S)$, s.t. $\sum_{x \in S} x = \sum_{x \in X \setminus S} x$?

Reduction:
We create a far division instance with two agents and identical valuations v. We set (with an abuse of notation) $\mathcal{G} = X$ and $v(x) = x$.
It is sufficient to notice that, for identical valuation, the more balanced the values of the two bundles the higher the Nash welfare. Therefore, denoted by $z = \sum_{i=1}^{t} x_i$, a partition exists if and only if the maximum Nash welfare is of $z/2$. ☐

Despite this negative result, it is still possible to compute an EFX allocation with good welfare guarantees. We next consider a greedy algorithm to compute an EFX allocation.

Greedy algorithm for identical valuations:

- $\mathcal{A} \leftarrow (\emptyset, \dots, \emptyset)$
- sort items $g_1, \dots, g_m$ in a non-increasing order, that is, $v(g_1) \geq v(g_2) \geq \ldots \geq v(g_m)$
- For $h = 1, \dots, m$
  - $i \leftarrow \arg\min_{i \in \mathcal{N}} v(A_i)$ (break ties arbitrarily)
  - $A_i \leftarrow A_i \cup \{g_h\}$

Theorem 12 (Barman et al. 2018)**.** *The greedy algorithm for identical valuations computes an* EFX *allocation which is an* 1.061 *approximation in terms of Nash social welfare.*

Proof. We only prove EFX. In particular, at every iteration EFX is satisfied.
At the beginning, the allocation is empty, and therefore, EFX.
Consider an arbitrary iteration where good g is allocated to, say, agent i. Before allocating g no one envies i since she has the minimum valued bundle and the allocation is EFX. For every $j, k \neq i$, after allocating g, the allocations remains EFX if i is not considered. Let us now consider i and observe that no j EFX-envies i since g is the minimum item in the new bundle (the algo introduces items in a non-increasing order) and before introducing g so one was envying i. On the other hand, also i does not EFX-envy any j, since her bundle has increased. □

6 Maximin-Share

The Maximin-Share is a relaxation of PROP. Motivated by the question of what can we guarantee in the worst case to the agents, the rationale of this concept is to think of a generalization of the well-known CutAndChoose protocol to multiple agents as follows:
Suppose that agent i is asked to partition the goods into n bundles and then the rest of the agents choose a bundle before i. In the worst case, agent i will be left with her least valuable bundle. Hence, a risk-averse agent would choose a partition that maximizes the minimum value of a bundle in the partition. This value is called the maximin share of agent i, and for $n = 2$, it is precisely what he could guarantee to himself in the discrete form of the CutAndChoose protocol, by being the cutter. The objective then is to find an allocation where every agent receives at least his maximin share.
Let us denote by $\Pi_n(\mathcal{G})$ the set of all possible allocations of goods in $\mathcal{G}$ among n agents.

Definition 11 (Maximin share)**.** *The maximin share of agent i is given by*

$$\mu_i = \mu_i(n, v_i, \mathcal{G}) = \max_{\mathcal{A} \in \Pi_n(\mathcal{G})} \min_j v_i(A_j) \,.$$

An allocation is MMS if every agent receives at least her maximin share.

Example 7. *Consider the instance in Table 1. The maximin shares of the agents are as follows: $\mu_1 = 6$, $\mu_2 = 7$, and $\mu_3 = 6$. The allocation $A_1 = \{g_4, g_5\}$, $A_2 = \{g_2, g_3\}$, $A_3 = \{g_1\}$ is MMS.*

Properties.

- $\mathsf{PS}_i \geq \mu_i$**.** In fact, PS_i is the average value of n bundles for i while μ_i is the minimum value of n bundles, for some specific allocation. Therefore, the inequality holds true.

- **Monotonicity.** $\mu_i(n-1, v_i, \mathcal{G} \setminus \{g\}) \geq \mu_i(n, v_i, \mathcal{G})$ for any $g \in \mathcal{G}$.
  Consider a partition of $\mathcal{G}$ that attains the maximin share of i. Let $\mathcal{A}$ be this partition and assume $g \in A_1$. Consider the remaining partition $(A_2, \ldots, A_n)$ and allocate goods in $A_1 \setminus \{g\}$ arbitrarily, obtaining a partition $(B_2, \ldots, B_n)$. This is a $(n-1)$-partition of $\mathcal{G} \setminus \{g\}$ where the value of agent i for any bundle is at least μ_i. Monotonicity follows.

6.1 Existence and Computation

Let us start by noticing that by definition, an MMS allocation always exists under identical valuations (the allocation which has minimum value μ_i is an MMS allocation).
Although the maximin share is a relaxation of the proportional share, it is still not guaranteed to exist. However, a good portion of work in fair division has focused on good approximations. An allocation $\mathcal{A}$ is an α approximation of MMS if for each i, $v_i(A_i) \geq \alpha \cdot \mu_i$. The best-known approximation so far is of $3/4 + 1/(12n)$.
We next present a simple algorithm achieving an $\frac{1}{2}$-approximation.

An $\frac{1}{2}$-approximation for MMS

- For each $i \in \mathcal{N}$ compute PS_i for the instance $\mathcal{N}, \mathcal{G}, \{v_i\}_{i \in \mathcal{N}}$
- while there exist i and g such that $v_i(g) \geq \frac{\mathsf{PS}_i}{2}$
  - assign g to i
  - $\mathcal{N} \leftarrow \mathcal{N} \setminus \{i\}$, $\mathcal{G} \leftarrow \mathcal{G} \setminus \{g\}$
  - update the proportional share of each remaining agent in the new instance $\mathcal{N}, \mathcal{G}, \{v_i\}_{i \in \mathcal{N}}$
- Run Round-Robin on the remaining instance $\mathcal{N}, \mathcal{G}, \{v_i\}_{i \in \mathcal{N}}$

Theorem 13 (Amanatidis et al. 2017)**.** *The algorithm outputs an $\frac{1}{2}$-approximation for MMS.*

Proof. Let $\mathcal{A}$ be the outcome of the approximation algorithm.
Consider an iteration k of the while loop. At this point, the algorithm has allocated $k-1$ items and $k-1$ agents have been removed. Let $\mathcal{N}$ and $\mathcal{G}$ be the initial set of goods and agents, and let $\mathcal{N}'$ and $\mathcal{G}'$ be the current set of goods and agents. Let i be the agent receiving the item g in the current iteration, agent i has a proportional share of PS'_i (the proportional share for $\mathcal{N}'$ and $\mathcal{G}'$).
Applying monotonicity $k-1$ times, we have $\mu_i(n-k+1, v_i, \mathcal{G}') \geq \mu_i(n, v_i, \mathcal{G})$; moreover, $\mathsf{PS}'_i \geq \mu_i(n-k+1, v_i, \mathcal{G}')$. Therefore i gets at least half of her true maximin share.
Let us now consider the agents involved in Round-Robin. Let $\mathcal{N}'', \mathcal{G}'', \{v_i\}_{i \in \mathcal{N}}$ be the instance at the beginning of Round-Robin. Notice that for no agent $i \in \mathcal{N}''$ there exists a good $g \in \mathcal{G}''$ such that $v_i(g) \geq \mathsf{PS}''_i/2$. Moreover, by monotonicity, we have $\mu_i(|\mathcal{N}''|, v_i, \mathcal{G}'') \geq \mu_i(n, v_i, \mathcal{G})$ for each i.
Round-Robin provides an $\mathsf{EF1}$ (and hence $\mathsf{PROP1}$) allocation for the instance $\mathcal{N}'', \mathcal{G}'', \{v_i\}_{i \in \mathcal{N}}$. Therefore, for any $i \in \mathcal{N}''$ there exists $g \in \mathcal{G}'' \setminus A_i$ such that

$$v_i(A_i) \geq \mathsf{PS}''_i - v_i(g) \geq \mathsf{PS}''_i/2 \geq \mu_i(|\mathcal{N}''|, v_i, \mathcal{G}'')/2 \geq \mu_i(n, v_i, \mathcal{G})/2\,,$$

and the statement follows. □

References

[1] Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Transactions on Algorithms (TALG)*, 13(4):1–28, 2017.

[2] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy algorithms for maximizing Nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 7–13, 2018.

[3] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[4] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.

[5] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 125–131, 2004.

[6] Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 675–692, 2014.