# On Fair Division under Heterogeneous Matroid Constraints

### Abstract

We study fair resource allocation among agents with additive valuations and matroid feasibility constraints. In these settings, every agent $i$ is associated with a matroid $\mathcal{M}_i$, and may receive only bundles that are independent in $\mathcal{M}_i$. Such scenarios have been of great interest in the AI community, as they encompass many real-world applications (e.g., allocation of shifts to medical doctors). A common fairness notion for indivisible goods is envy-free up to 1 good (EF1), which is a natural relaxation of envy freeness. Previous studies imply EF1 solutions for either complete allocations (where all goods are allocated) with homogeneous agents, or partial allocations with heterogeneous agents, where heterogeneity may refer to the agents' feasibility constraints or their valuations. A major open problem is the existence of fair allocations in settings with both complete allocations and heterogeneous agents. In this work, we make several steps towards resolving this problem. For settings with heterogeneous constraints, we use the notion of feasible EF1 (F-EF1), which captures envy under feasibility constraints. We establish positive and negative results for the existence of F-EF1 in various settings with heterogeneous agents, including different matroid types and different valuation types.

## 1 Introduction

The problem of fair division of indivisible goods has attracted a large body of recent work in artificial intelligence and algorithmic game theory literature. See Brandt et al. (2016); Endriss (2017) and references therein for comprehensive surveys. As many applications concern the division of indivisible goods, and algorithms for fair division of divisible goods are often inapplicable to such settings, algorithms for this domain are crucially desired. This problem is not a mere theoretical exercise: it arises in many real-life settings of resource allocations, such as the ones implemented in the Spliddit website (Goldman and Procaccia 2015) algorithms for fairly dividing course seats among students in various universities (Budish 2011).

In a fair division problem with additive valuations, a set $M$ of $m$ indivisible goods should be allocated among a set $N$ of $n$ agents. Every agent $i$ has a valuation function $v_i$ which assigns a real value $v_{ij}$ to every item $j$, so that agent $i$'s value for a set of items $S$ is $v_i(S) = \sum_{j \in S} v_{ij}$. An allocation is a partition of the set of items among the agents; it is denoted by a vector $X = (X_1, \ldots, X_n)$, where $X_i \subseteq M$ for every agent $i$ and $X_i \cap X_j = \emptyset$ for every $i, j \in N$.

A common notion of fairness is that of *envy freeness (EF)*, which means that every agent (weakly) prefers her bundle to the bundle of any other agent: for every $i, j \in N$, $v_i(X_i) \geq v_i(X_j)$. While EF is a powerful notion in applications with divisible goods, it may be unattainable for indivisible goods.

In this paper we focus on a very common relaxation of EF, denoted *EF1 — envy free up to one good* (Budish 2011) . An allocation $X$ is EF1 if every agent $i$ (weakly) prefers her bundle to any other agent $j$'s bundle, up to the removal of the best good (in $i$'s eyes) from agent $j$'s bundle. I.e., for every two agents $i, j$, if $X_j \neq \emptyset$, then there exists an item $g \in X_j$ such that $v_i(X_i) \geq v_i(X_j \setminus \{g\})$. An EF1 allocation always exists and can be computed efficiently (Lipton et al. 2004). Moreover, there always exists an allocation that is both EF1 and *Pareto efficient — no other allocation is at least as good for everyone and strictly better for someone* (Caragiannis et al. 2019). In a sense, EF1 has become the analog of EF for settings with indivisible items.

(**Erel:** I think it is important to put our main focus — different constraints — already at the first page.)

However, these strong results apply only in unconstrained settings, in which any partition of the goods among agents is allowed. In many settings of interest, the set of possible allocations is inherently constrained. Moreover, in general, *different agents may have different constraints*. For example, consider the allocation of employees among departments of a company: one department has room for 4 project managers and 2 backend engineers, while another one may need department B has room for 3 backend engineers and 5 data scientists, etc. Another example is assigning shifts to shift workers, where every worker has their own schedule limitations. Such real-life constraints can be modeled by *matroids* (see Definition 2.1), particularly *partition matroids*. In a partition matroid, the set of items is partitioned into a set of *categories*, and every category is associated with a *cap* on the number of goods from that category that can be allocated to the agent. There are two ways to address such constraints.

The first approach is to directly construct allocations that satisfy the constraints, i.e., guarnatee that each agent receives a feasible bundle. This approach was recently taken by Biswas and Barman (2018, 2019). Assuming that at least one feasible allocation exists, they establish the existence of EF1 allocations in scenarios where agents have: (i) identical matroid constraints and identical valuations, and (ii) identical partition matroid constraints, even under heterogeneous valuations (see Section 2 for details). However, their algorithms cannot handle agents with different partition constraints (Section 3) or general matroid constraints with different valuations (Section ???).

A second approach is to treat the constraints as a part of the valuation function. That is, the valuation of an agent to a bundle equals the value of the best feasible subset of the bundle. This approach seemlessly handles heterogeneity in both constraints and valuations. The valuation functions constructed this way are no longer additive, but they are *submodular*. Recently, Babaioff, Ezra, and Feige (2020). and Benabbou et al. (2020) have independently proved the existence of EF1 allocations when agents have submodular valuations with binary marginals. Such an allocation can be converted to a fair *and feasible* allocation by giving to each agent the best feasible subset of his allocated bundle, and disposing the other items. However, this is possible only when there is *free disposal*, which is not always the case. For example, when allocating shifts to doctors, if an allocation rule returns an unfeasible allocation and we dispose shifts to make it feasible, the emergency-room might remain empty for a shift. A similar situation in the context of allocating papers to referees may leave some papers without reviews. The allocation rules developed in the above paper do not work when they are constrained to return feasible allocationsm, as we show in Section 3. Thus, a major open question remains:

> ***Open problem**. Given agents with different additive valuations and different matroid constraints, does there always exist a complete and feasible EF1 allocation?*

## 1.1 Contribution and Techniques

This paper makes some steps towards solving this problem.

**Impossibilities.** We first set the boundaries to our exploration by observing some simple impossibility results.

First, consider a setting with 2 agents and 8 identical items of a single category, where Alice's capacity is 3 Bob's capacity is 5. The only complete feasible allocation gives 3 items to Alice and 5 to Bob. By mere valuation considerations, it is not EF1, since even after removing a single item from Bob's bundle, Alice values it at 4, which is more than her value for her own bundle. However, a bundle of 4 items is infeasible for Alice. Therefore, a more reasonable definition of envy in this setting is *feasible envy*, in which each agent compares her bundle against the best feasible subset of any other agent's bundle (see Section 2 for the formal definition). In the example above, the best feasible subset of Bob's bundle for Alice is worth 3, thus the allocation is *feasibly-envy-free (F-EF)*. If Alice values one of Bob's items at 2, then the above allocation is not F-EF since the best feasible subset of Bob's bundle for Alice is worth 4. But it is *F-EF1*, since it becomes F-EF after removing this item from Bob's bundle. Note that F-EF1 is equivalent to EF1 when agents have identical constraints.

Second, we show that if the partition of items into categories is different for different agents, an F-EF1 allocation may not exist, even for two agents with identical valuations (see Example 3.5).

Third, we show that going beyond matroid constraints to *graph-matching constraints* (an intersection of two matroids) is hopeless: even with 2 agents with identical valuations and identical matching constraints, an EF1 allocation may not exist (Example 3.4).

Fourth, going beyond EF1 to the stronger notion of *envy-free up to any good (EFX)* is hopeless: even with 2 agents with identical valuations and identical matroid constrains, an EFX allocation may not exist (Example 3.3). (**Erel:** Do we have an non-existence example for EFX with partition matroids?)

Based on these results, we focus on finding F-EF1 allocations when the agents' constraints are represented by either (1) *partition matroids* where all agents share the same partition of items into categories but may have different capacities; (2) *general matroids* where all agents have the same matroid constraints but may have different valuations.

**Algorithms** For partition matroids, we identify the reason for which the algorithms of Lipton et al. (2004) and Biswas and Barman (2018) fail for agents with different capacities: it is the process of *cycle-removal*. Informally (see Section 2 for details), these algorithms maintain a directed *envy-graph* in which each agent points to the agents he envies. The algorithm prioritizes the agents who are not envied, since giving an item to such agents keeps the allocation EF1. If there are no unenvied agents, then there must be a cycle in the envy-graph, and it can be removed by exchanging bundles among the involved agents. This process does not work when different agents may have different constraints (See example XX). Our main challenge is thus to develop techniques that guarantee that no envy-cycle is created in the first place. We manage to do this in four different settings:

1. All agents have *binary valuations* — the value of each item is either 0 or 1 **(Section ...)**.
2. All agents have *identical* valuations **(Section ...)**.
3. There are *two agents* **(Section ...)**.
4. There are at most *two categories* **(Section ...)**.

Each setting is solved by a different algorithm and using a different cycle-prevention technique.

For *general matroid constraints*, we present algorithm that can handle agents with different additive valuations in the following cases **(Section ...)**:

1. At most two agents;
2. At most three agents with binary valuations.

(**Erel:** Can we start the enumeration at 5?)

Table 1 compares our results with known results.

| Matroid Type | Complete Allocation | Heterogeneity in Constraints | Heterogeneity in Valuations | Binary / General | # of agents | Remark | Source / Section |
|---|---|---|---|---|---|---|---|
| Uniform | V | V | V | General | $n$ | | Section 4.1 |
| Partition | V | - | V | General | $n$ | | Biswas and Barman (2018) |
| | V | V | - | General | $n$ | | Section 4.2 |
| | V | V | V | General | 2 | | Section 6.2 |
| | V | V | V | Binary | $n$ | Pareto-efficient if caps in $\{0, 1\}$ | Section 5 |
| General | V | - | - | General | $n$ | | Biswas and Barman (2018) |
| | - | V | V | Binary | $n$ | Pareto efficient | Babaioff, Ezra, and Feige (2020)<br>Benabbou et al. (2020) *** |
| | V | - | V | Binary | 3 | Pareto efficient | Section 7 |
| | V | - | V | General | 2 | | Section 7 |
| | V | V | V | General | 2 | *Non-existence* | Example 3.5 |

Table 1: A summary of our results in the context of previous results. All results are for additive valuation functions. The result marked by *** is not mentioned explicitly in the references, but it follows from them by disposing items (see Introduction). (**Erel:** Where is the result on 2 categories?)

### 1.2 Related Work

Capacity constraints are common in matching markets such as doctors–hospitals and workers–firms; see Klaus, Manlove, and Rossi (2016) for a recent review. In these settings, the preferences are usually represented by ordinal rankings rather than by utility functions, and the common design goals are Pareto-efficiency, stability and strategyproofness, rather than fairness.

Fair allocation with capacity constraints is particularly relevant to the problem of *assigning conference papers to referees*. Garg et al. (2010); Long et al. (2013); Lian et al. (2017) study a setting in which for each agent (reviewer) there is both an upper and a lower capacity on the total number of items. The constraints may be different for each agent, but there is only one category of items. Note that lower capacities are not matroid constraints, since they are not downwards-closed. The same is true in the setting studied by Ferraioli, Gourvès, and Monnot (2014), where each agent must receive exactly $k$ items.

Fair allocation of items of different categories has been studied by Mackin and Xia (2016); Sikdar, Adali, and Xia (2017). There are $k$ categories, each of which has $n$ items, and each agent must receive exactly one item of each category. Sikdar, Adalı, and Xia (2019) consider an exchange market in which each agent holds multiple items of each category and should receive a bundle with exactly the same number of items of each category. Nyman, Su, and Zerbib (2020) study a similar setting (they call the categories “houses” and the objects “rooms”), but with monetary transfers (which they call “rent”).

Barrera et al. (2015); Bilò et al. (2018); Suksompong (2019) study another kind of constraints in fair allocation. The goods are arranged on a line, and each agent must receive a connected subset of the line (as when each item is a house and each agent should get a connected part of the street). Bouveret et al. (2017); Bei et al. (2019) study a more general setting in which the goods are arranged on a general graph, and each agent must receive a connected subgraph. Note that these are not matroid constraints.

Gourvès, Monnot, and Tlilane (2013) study a setting with a single matroid, where the goal is to build a base of the matroid and provide worst case guarantees on the agents’ utilities. Gourvès, Monnot, and Tlilane (2014) and Gourvès and Monnot (2019) require the *union* of bundles allocated to all agents to be an independent set of the matroid. This inherently implies a free-disposal assumption, which we do not make here.

Fair allocation with *binary additive valuations* (without constraints) has been studied recently, due to its practical applications (Aleksandrov et al. 2015). Binary valuations allow to attain better fairness guarantees (Bouveret and Lemaître 2016; Barman et al. 2017; Amanatidis et al. 2020) and mechanisms with better strategic properties (Halpern et al. 2020). While in general the MNW solution is NP-hard, with binary valuations it can be computed efficiently (Darmann and Schauer 2015; Barman, Krishnamurthy, and Vaish 2018).

## 2 Model and Preliminaries

### 2.1 Model

We consider settings where a set $M$ of $m$ items should be allocated among a set $N$ of $n$ agents. Every agent $i$ is associated with an *additive* valuation function $v_i$, which assigns a real value to every subset of items $S \subseteq M$. For ease of notation, we use $v_i(g) := v_i(\{g\})$ for any single item $g$. A valuation $v_i$ is *additive* if $v_i(S) = \sum_{j \in S} v_i(j)$ for every $S \subseteq M$.

A valuation $v_i$ is *binary* if $v_i(j) \in \{0, 1\}$ for every $i \in N, j \in M$.

An allocation is denoted by $X = (X_1, \dots, X_n)$, where $X_i \subseteq M$ is the bundle given to agent $i$, and $X_i \cap X_j = \emptyset$ for all $i, j \in N$. An allocation is *complete* if $\bigcup_{i \in N} X_i = M$.

In our setting, every agent $i$ is associated with a matroid $\mathcal{M}_i = (M, \mathcal{I}_i)$ that specifies the feasible bundles for agent $i$.

**Definition 2.1.** A pair $\mathcal{M}_i = (M, \mathcal{I})$ of a set of elements $M$ and a set of subsets $\mathcal{I} \subseteq P(M)$ (termed the set of independent sets) is a *matroid* if it satisfies the following properties:

- $\emptyset \in \mathcal{I}$
- For every $S, S' \subseteq S$, if $S \in \mathcal{I}$, then $S' \in \mathcal{I}$.
- For every $S, T \in \mathcal{I}$, if $|S| > |T|$, then there exists $g \in S \setminus T$ such that $T \cup \{g\} \in \mathcal{I}$.

We next define the notion of feasible allocations.

**Definition 2.2.** (feasible allocation) An allocation $X$ is said to be *feasible* if:

- it is individually feasible: $X_i \in \mathcal{I}_i$ for every agent $i$, and
- it is complete: $\biguplus_i X_i = M$

That is, an allocation is feasible if every agent receives a feasible bundle according to her matroid, and all items are allocated. The set of all feasible allocations is denoted by $\mathcal{F}$.

Throughout this paper, we restrict attention to instances that admit a feasible allocation.

**Assumption 2.3.** *There exists at least one feasible allocation: $\mathcal{F} \neq \emptyset$.*

We next define a special case of matroid, called partition matroid.

**Definition 2.4.** (partition matroid) A matroid $\mathcal{M}_i = (M, \mathcal{I}_i)$ is a *partition matroid* if

- for some $\ell_i \leq m$, there exists a set of subsets of $M$, $C_i^1, \ldots, C_i^{\ell_i}$, that form a partition of $M$. These sets are termed *categories*,
- every category $C_i^h$, $h \in [\ell_i]$, is associated with some capacity $k_i^h$, and
- The collection of independent sets is

$$\mathcal{I}_i = \{S \subseteq M \mid |S \cap C_i^h| \leq k_i^h \text{ for every } h \in [\ell_i]\}.$$

A special case of a partition matroid is a *uniform matroid*:

**Definition 2.5.** (uniform matroid) A matroid $\mathcal{M}_i = (M, \mathcal{I}_i)$ is a *uniform matroid* if $\mathcal{I}_i = \{S \subseteq M : |S| \leq k_i\}$ for some capacity $k_i$.

An instance with partition matroids is said to have *identical categories* if all the agents have the same categories, but not necessarily the same capacities. I.e., $\ell_i = \ell_j = \ell$ for every $i, j \in N$, and $C_i^h = C_j^h = C^h$ for every $h \in \ell$.

An instance is said to have *identical matroids* if all agents have the same matroid feasibility constraints. I.e., $\mathcal{I}_i = \mathcal{I}_j$ for all $i, j \in N$.

## 2.2 Fairness Notions

In this section we present several definitions of fairness regarding allocations of indivisible goods.

**Definition 2.6** (envy and envy freeness)**.** Given a feasible allocation $X$, agent $i$ **envies** agent $j$ iff $v_i(X_i) < v_i(X_j)$. $X$ is *envy free* iff no agent envies another agent.

**Definition 2.7** (EF1)**.** An allocation $X$ is *envy free up to 1 good (EF1)* iff for every $i, j \in N$, if $X_j \neq \emptyset$, then there exists $g \in X_j$ such that $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

**Definition 2.8** (best feasible subset)**.** The *best feasible subset* of a set $S$ for agent $i$ is

$$\textsc{Best}_i(S) := \operatorname*{argmax}_{T \subseteq S,\ T \in \mathcal{I}_i} v_i(T).$$

Note that $\textsc{Best}_i(S)$ may not be uniquely defined. When clear in the context, we abuse notation and use $\textsc{Best}_i(S)$ as an arbitrary set in $\operatorname{argmax}_{T \subseteq S,\ T \in \mathcal{I}_i} v_i(T)$.

In the case of partition matroids, the best feasible set decomposes into the different categories; i.e.,

$$\textsc{Best}_i(S) = \biguplus_{h \in [\ell_i]} \textsc{Best}_i(S^h), \text{ where } S^h = S \cap C_i^h. \quad (1)$$

**Definition 2.9** (feasible valuation)**.** Given a feasibility constraint, the *feasible valuation* of agent $i$ for a set $S$ is

$$\hat{v}_i(S) = v_i(\textsc{Best}_i(S)).$$

**Definition 2.10.** Given a feasible allocation $X$:

- Agent $i$ *F-envies* agent $j$ iff $\hat{v}_i(X_i) < \hat{v}_i(X_j)$.
- $X$ is *F-EF* feasible envy-free) if no agent F-envies another agent.
- $X$ is *F-EF1* iff for every $i, j \in N$:

$$\text{if } X_j \neq \emptyset, \text{then } \exists g \in X_j \text{ s.t. } v_i(X_i) \geq \hat{v}_i(X_j \setminus \{g\}).$$

Another usfull notation we use is *positive envy*, which is, intuitively, the amount by which agent $i$ envies some other agent $j$.

**Definition 2.11.** Given an allocation $X$, the *positive envy* of agent $i$ towards agent $j$ is:

$$\mathit{Envy}^+(i, j) := \max(0, \hat{v}_i(X_j) - \hat{v}_i(X_i))$$

## 2.3 Common Tools and Techniques

Below we recall the most common methods for finding an EF1 allocation.

**Envy Cycles Elimination** The first method for attaining an EF1 allocation is due to Lipton et al. (2004).

**Definition 2.12.** The *envy graph* of an allocation $X$, $G(x)$, is a directed graph where the nodes represent the agents, and there is an edge from agent $i$ to agent $j$ iff $v_i(X_i) < v_i(X_j)$

The *envy cycles elimination* algorithm works as follows: At every step, choose an agent that has no incoming edges in the envy graph. Give this agent an arbitrary unallocated item, and re-draw the envy graph. If there is a directed cycle in the graph, switch the agents' bundles along the cycle. Do so until no cycles remain (guaranteeing the existence of an agent with no incoming edges), and continue to the next step. This procedure was essentially proved to result in an EF1 allocation (the term EF1 was not invented, but it was proved to hold equivalent traits) for the unconstrained setting even under general valuations (Lipton et al. 2004).

We use the term *feasible envy graph* to indicate an envy graph created by the feasible-envy instead of plain envy.

**Max Nash Welfare** The first proof of existence of EF1 allocations that are also *Pareto efficient* is due to Caragiannis et al. (2019).

**Definition 2.13.** Given an allocation $X$:

- The *Nash Social Welfare* (NSW) of $X$ is the product of the agents' values: $\prod_{i \in [n]} v_i(X_i)$
- $X$ is *Maximum Nash Welfare* (MNW) if it maximizes NSW among all feasible allocations.

(**Amitay:** TODO move to intro) Maximum Nash welfare (MNW) allocations have been recently seen as a way to unite all fairness criteria into one notion, and indeed they seem to accomplish several fairness criteria at once in different settings. In particular, for additive valuations a MNW allocation is proved to be EF1. (Caragiannis et al. 2019).

**Round Robin** A common simple algorithm for fair allocation is the *round robin* procedure. Given a set of items, a set of agents and an order $\sigma$ of the agents, the algorithm works as follows: While there are still unallocated items, give the next agent in $\sigma$ (or the first agent if reaching the end of $\sigma$) the item she values most among the unallocated items. Simple as it might be, this algorithm results in an EF1 allocation in the unconstrained setting for additive valuations. (Caragiannis et al. 2019)

**Round Robin per Category + Envy Cycle Elimination** Biswas and Barman (2018) suggested an algorithm (algorithm 1) for homogenous partition constraints. Their algorithm runs round-robin on each category in turn, where the picking order in each category is determined by a topological order on the envy-graph, such that each agent picks an item before all agents she envies. They prove the resulting allocation is EF1 for the homogenous partition setting.

**Algorithm 1:** Per-Category Round Robin, (Biswas and Barman 2018)

```
initialize:
$\sigma \leftarrow$ an arbitrary order over the agents.
$\forall i \in [n]$ $X_i \leftarrow \emptyset$
for every category $h$ do
    Run round robin with $C^h, \sigma$;
    Let $X_i^h$ be the resulting allocation for agent $i$;
    $\forall i \in [n]$ $X_i \leftarrow X_i \cup X_i^h$;
    Draw envy graph for current allocation;
    Remove cycles from the graph, switching bundles
      along the cycles;
    Set $\sigma$ to be a topological order of the graph;
end
```

# 3 Negative Results and Failed Attempts

In this section we show some negative results and give intuition for why common approaches fail in settings with heterogeneous constraints.

## 3.1 Partition Matroids

**Round Robin Per Category + Envy Cycle Elimination** We next observe that algorithm 1 may fail in the heterogeneous constraints setting.

| Category | Capacities | Alice | Bob |
|---|---|---|---|
| $C^1$ | $k_A^1 = 1$<br>$k_B^1 = 1$ | 1,1 | 1,0 |
| $C^2$ | $k_A^2 = 1$<br>$k_B^2 = 0$ | 0,0 | ▨ |
| $C^3$ | $k_A^3 = 0$<br>$k_B^3 = 1$ | ▨ | 0,0 |
| $C^4$ | $k_A^4 = 1$<br>$k_B^4 = 1$ | 0,1 | 1,1 |

Table 2: Possible Intermediate allocation of algorithm 1 applied on heterogeneous capacities may contain an envy cycle that cannot be removed by bundle switching.

**Example 3.1.** Consider an instance with 4 categories and 2 agents, and for simplicity, mark the valuations of items as an ordered pair $v_{Alice}, v_{Bob}$ (e.g. an item $0, 1$ is an item Alice values $0$ and Bob values $1$). The allocation described in table 2 can be the intermediate outcome of algorithm 1 before the stage of removing envy cycles after category 4. If Alice begins and the choosing order is: $C^1$: A:$1, 1$, B:$0, 1$; $C^2$: A:$0, 0$; $C^3$: B:$0, 0$; $C^4$: B:$1, 1$, A:$0, 1$; This intermediate allocation contains a cycle in the envy graph. Algorithm 1 removes cycles by switching bundles, but here it is impossible due to different feasibility constraints in $C^2$ and $C^3$. Without removing the cycle, even one extra item added to any of their bundles might cause a non-EF1 allocation.

**Maximum Nash welfare** Next, we observe that even with homogeneous constraints, MNW does not imply EF1 even with binary valuations. (**Amitay:** this did appear in a footnote in (Biswas and Barman 2018). not an example but the existence of such an example. should we remove it or add a comment?)

**Example 3.2.** Consider the non-EF1 allocation in Table 3. The total values of the agents in this allocation are $v_A(X_A) = 2, v_B(X_B) = 3$, resulting in Nash welfare of 6. The only other possible value profiles are $(1, 4)$ or $(0, 5)$, resulting in lower Nash welfare. Hence, the allocation is MNW.

Note that Benabbou et al. (2020) prove that MNW always implies EF1 for submodular valuations with binary marginals. However, they consider only *clean* allocations, where items with $0$ marginal value are not allocated. In contrast, we require that all items be allocated.

In order to decide where to focus our attention, we considered several settings and fairness criteria to get a sense of what settings may admit a fair allocation and which criteria might be achievable.

## 3.2 General Matroids, Non existence of EFX

(**Erel:** Should we consider F-EFX?) (**Amitay:** since this example if with identical matroids, EFX and F-EFX are the

| Category | Alice | Bob |
|---|---|---|
| $C^1$ | 1,1 | 0,0 |
| $k_A^1 = k_B^1 = 2$ | 1,1 | 0,0 |
| $C^2$ | 0,1 | 0,1 |
| $k_A^2 = k_B^2 = 3$ | 0,1 | 0,1 |
|  | 0,1 | 0,1 |

Table 3: Example where for different binary valuations, even with the same partition matroid, a MNW allocation is not EF1. Also, round robin might fail at achieving EF1. The table presents the MNW allocation, that can also be the result of running round robin, and shows for every item the valuations of the agents for it (for example, for item $g = 0, 1$ the values are $v_A(g) = 0, v_B(g) = 1$)

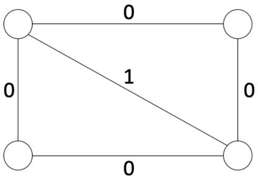

Figure 1: Identical valuations, identical matroids, no feasible EFX allocation. This graphic matroid, with the valuation indicated on the edges, admits no EFX allocation for 2 agents.

same, as any feasible bundle is feasible for every agent)

An *Envy Free up to any good* (EFX) allocation is a feasible allocation $X$ where for every pair of agents $i, j$, for every good $g$ in $j$'s bundle, $v_i(X_i) \geq v_i(X_j \setminus \{g\})$. It is a stronger fairness notion than EF1, not proved to exist even for the unconstrained setting. We show that in the constrained setting, even with homogeneous constraints and homogeneous agents such allocation is not guaranteed to exist.

We will use a *graphic matroid* for the example. A graphic matroid is represented by a graph where the edges correspond to items, and a set of items is an independent set if the corresponding edges do not form a cycle.

**Example 3.3.** Even under identical matroids and identical binary valuations with 2 agents, a complete EFX allocation may not exist. Consider the graphic matroid in figure 1, with the valuation indicated on the edges. For 2 agents, the only EFX allocation gives the diagonal edge as a single item bundle to one agent, and the rest of the edges to the other. The second bundle creates a cycle in the graph, so it is not a feasible allocation.

### 3.3 Matching Constraints, Non existence of EF1

We also tried to extend beyond matroid constraints, but even for the slight extension to *matching constraints*, we found an EF1 allocation may not exist. Matching constraints require that each bundle be a matching in a given bipartite graph (the edges are the items). Note that the set of possible matchings is not a matroid, but it is an intersection of two matroids.

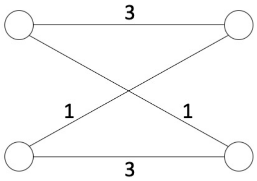

Figure 2: Example- Matching Constraints with no feasible EF1 allocation. The only feasible allocation gives one agent the diagonal edges and the other one the horizontal edges, and it is not EF1.

**Example 3.4.** In the bipartite graph described in figure 2, there are only 2 possible matchings: the diagonal edges and the horizontal edges. For two agents with identical valuations (indicated in the figure over each edge), the allocation that gives one of them the horizontal edges and the other the diagonal ones is not EF1, as the second agent would envy the first even after removing any of the items in her bundle.

### 3.4 Partition with Different Categories, Non existence of F-EF1

In the case of general partition matroids, even where valuations are identical and there are only 2 agents, there exists an instance that does not admit an F-EF1 allocation.

**Example 3.5** (Different partition, no F-EF1)**.** Let $M = \{a, b, c, d\}$, $N = \{1, 2\}$, $v_1 = v_2 = v$, $v(a) = v(b) = 10$ and $v(c) = v(d) = 1$. Let the partitions for agent 1 be: $C_1^1 = \{a, c\}, C_1^2 = \{b, d\}$ with capacities: $k_1^1 = k_1^2 = 1$ and for agent 2: $C_2^1 = \{a\}, C_2^2 = \{b\}, C_2^3 = \{c, d\}$ with capacities $k_2^1 = k_2^2 = 1, k_2^3 = 0$:

| Item: | a | b | c | d |
|---|---|---|---|---|
| $v_1 = v_2 = v$ : | 10 | 10 | 1 | 1 |
| Category for agent 1: | 1 | 2 | 1 | 2 |
| Category for agent 2: | 1 | 2 | 3 | 3 |

The only feasible allocation is $X_1 = \{c, d\}, X_2 = \{a, b\}$. and it is not F-EF1 for agent 1.

Therefore, in the remainder of this paper when considering partition matroid we restrict attention to identical partitions (but, possibly different capacities and valuations).

## 4 Warm-up: Uniform Matroids and Partition Matroids with Identical Valuations

The main question in this section is under what conditions an F-EF1 allocation is guaranteed to exist.

(Biswas and Barman 2018) proved existence for identical capacities and additive valuations:

**Theorem 4.1** (Biswas and Barman)**.** *For Identical Partition matroid with Identical Capacities, there always exists an EF1 allocation.*

### 4.1 Uniform Matroid, Different Capacities, Different Valuations

As a warm-up, we consider a partition matroid with a single category, also known as a *uniform matroid*. We show an algorithm that finds an F-EF1 allocation for $n$ agents with different valuations and different capacities. This algorithm (Algorithm 2) is a slight modification of round robin where if an agent reached her capacity she is being skipped over. It will serve as a sub-routine in the following sections.

**Theorem 4.2.** *For a uniform matroid, a F-EF1 allocation always exists and can be found by Capped Round Robin (Algorithm 2). Furthermore, if $X$ is the resulting allocation, for every $i, j$ such that $i$ precedes $j$ in $\sigma$ , $v_i(X_i) = \hat{v}_i(X_i) \geq \hat{v}_i(X_j)$*

The proof for this is similar to that of the unconstrained setting and standard round robin. Note, that it does not work for more than one category, as can be seen in section 3.

**Algorithm 2:** Capped Round Robin

```
Input: Category $h$, capacities $k_i^h$ for every $i \in [n]$,
       and an order $\sigma$ of $[n]$
Initialize:
$L \leftarrow C^h, P \leftarrow \{i : k_i^h = 0\}, t \leftarrow 0,$
 $\forall i \in [n]\ X_i^h \leftarrow \emptyset$
while $L \neq \emptyset$ do
    $i \leftarrow \sigma[t]$;
    if $i \notin P$ then
        $g = \text{argmax}_{g \in L}\, v_i(\{g\})$;
        $X_i^h \leftarrow X_i^h \cup \{g\}$; {Agent $i$ gets her best
         unallocated item in $C^h$}
        $L \leftarrow L \setminus \{g\}$;
        if $|X_i^h| == k_i^h$ then
            $P \leftarrow P \cup \{i\}$ {Agent $i$ cannot get any
             more items from $C^h$}
        end
    end
    $t \leftarrow t + 1 \mod n$;
end
Return $X^h$
```

## 4.2 Different Capacities, Identical Valuations

We now consider an arbitrary number of categories, allow agents to have different capacities, but assume that all agents have the same valuations; this is a "dual" of the setting of Biswas and Barman (2018). Note that in this setting, the algorithm we used for a uniform matroid might fail.

**Lemma 4.3.** *In the setting of identical additive valuations and partition matroids with identical categories (but possibly different capacities), it is an invariant that throughout the stages of Algorithm 3 at no points are there cycles in the envy graph. (**Amitay:** Possible rephrasing: For identical valuations and different capacities, Algorithm 3 creates no cycles in the feasible envy graph at any stage. )*

*Proof.* Assume towards contradiction that at some point while running Algorithm 3 there is a cycle in the feasible envy graph. Denote the agents in the cycle $1 \dots p$, ordered by the order of the cycle. The existence of the cycle implies:

$$v(X_1) < \hat{v}_1(X_2) \leq v(X_2) < \hat{v}_2(X_3) \\ \leq v(X_3) < \dots \leq v(X_p) < \hat{v}_p(X_1) \leq v(X_1),$$

a contradiction. □

**Theorem 4.4.** *Every instance with identical additive valuations and partition matroids with identical categories (but different capacities) admits an F-EF1 allocation. Furthermore, Algorithm 3 finds such allocation.*

*Proof.* We show by induction that after every category the allocation is F-EF1. Induction base: after the first category the allocation is F-EF1 according to Lemma 4.2. Induction step: assume the allocation is F-EF1 after $t$ categories. Before running category $t + 1$ we reorder the agents topologically according to the feasible envy graph, and use this order as $\sigma$ in Algorithm 2. this is possible by Lemma 4.3. For every $i, j$ such that $i$ precedes $j$ in $\sigma$, $j$ does not F-envy $i$. By Lemma 4.2, during category $t + 1$'s iteration $j$ can become envious of $i$, but only up to F-EF1. TODO re-phrase: Also by 4.2, during category $t + 1$'s iteration $i$ cannot gain any feasible envy towards $j$, and since the allocation before the $t + 1$'th category was F-EF1, it remains F-EF1 after it. □

**Algorithm 3:** Per-Category Capped Round Robin

```
Input: $M, C, k_i^h$ for every $i \in [n], h \in [l]$
Result: an allocation $X$ which is $F - EF1$
initialize:
$\sigma \leftarrow$ an arbitrary order over the agents.
$\forall i \in [n]\ X_i \leftarrow \emptyset$
foreach $C^h \in C$ do
    Run Capped Round Robin with $C^h, \sigma$;
    $\forall i \in [n]\ X_i \leftarrow X_i \cup X_i^h$;
    Draw "Feasible Envy" directed graph where
     $(i, j) \in E \iff v_i(X_i) < \hat{v}_i(X_j)$;
    Set $\sigma$ to be a topological order of the graph
     (Lemma 4.3 claims it is a-cyclic);
end
```

We next show that for different capacities and identical valuations Maximum Nash Welfare implies F-EF1.

**Theorem 4.5.** *For different capacities and identical valuations, any allocation that maximizes Nash Social Welfare is F-EF1.*

*Proof.* Let $X$ be an allocation that maximizes Nash social welfare (MNW), and suppose there exist agents $i, j$ such that $i$ f-envies $j$.

Case 1: there exist a category $h$ such that $|X_i^h| < k_i^h$ and $\hat{v}_i(X_j^h) > 0$. In this case, the original proof for the non-constrained setting shall work: Let $g \in X_j^h$ such that $v(g) > 0$. Since $|X_i^h| < k_i^h$, $X_i \cup \{g\}$ is a feasible allocation. $X$ being MNW implies:

$$(v(X_i) + v(g)) \cdot (v(X_j) - v(g)) \leq v(X_i) \cdot v(X_j)$$

opening the parenthesis:

$$v(X_i)v(X_j)-v(X_i)v(g)+v(X_j)v(g)-v(g)^2 \leq v(X_i)v(X_j)$$

Simplifying the above expression and using the fact that $v(g) > 0$ we get:

$$v(X_j) - v(g) \leq v(X_i)$$

Since valuations are additive, this implies

$$v(X_j \setminus \{g\}) = v(X_j) - v(g) \leq v(X_i)$$

By the fact that $\hat{v}_i(S) \leq v(s)\ \forall S$, we get the desired feasible EF1 property:

$$\hat{v}_i(X_j \setminus \{g\}) \leq v(X_j \setminus \{g\}) \leq v(X_i) = \hat{v}_i(X_i)$$

Case 2: for every category $h'$ such that $|X_j^{h'}| > 0$, it holds that $|X_i^{h'}| = k_i^{h'}$. Let $h$ be some category in which $i$ feasibly envies $j$. that is: $v(X_i^h) = \hat{v}_i(X_i^h) < \hat{v}_i(X_j^h)$. Such a category exists since otherwise $i$ wouldn't be able to envy $j$ feasibly at all. Let $g = \text{argmax}_{t \in X_j^h} v(t)$, $b = \text{argmin}_{t \in X_i^h} v(t)$. Such items exist because $|X_i^h| = k_i^h > 0$ otherwise $i$ wouldn't have feasible envy for $j$ in that category, and $|X_j^h| > 0$ otherwise $i$ wouldn't envy $j$ at all at that category. Than $v(X_i^h) \geq k_i^h \cdot v(b)$ and $v(X_j^h) \leq k_j^h \cdot v(g)$. We know that $i$ envies $j$ a feasible envy in $C^h$ so: $k_i^h \cdot v(g) \geq \hat{v}_i(X_j^h) > \hat{v}_i(X_i^h) = v(X_i^h) \geq k_i^h \cdot v(b)$ therefore $v(g) - v(b) > 0$.

We will use a hypothetical item exchange for the proof: look at the allocation that exchanges the items $b, g$. It is feasible since they are in the same category. We know that $X$ is MNW, so the value product of $i, j$ should be higher than in the new allocation:

$$(v(X_i)+v(g)-v(b))\cdot(v(X_j)-v(g)+v(b)) \leq v(X_i)\cdot v(X_j)$$

let $z = v(g) - v(b)$. We get:

$$v(X_i)v(X_j) - v(X_i)z + v(X_j)z - z^2 \leq v(X_i)v(X_j)$$

Simplifying the above expression and using the fact that $z > 0$, we get:

$$v(X_j) - z \leq v(X_i)$$

Since the valuation is additive, and $v(b) \geq 0$, we get:

$$v(X_i) \geq v(X_j) - z = v(X_j) - v(g) + v(b)$$
$$\geq v(X_j) - v(g) = v(X_j \setminus \{g\})$$

Because the value for $i$'s best feasible subset of $X_j$ is smaller or equal to $v(X_j)$, we get that:

$$\hat{v}_i(X_j \setminus \{g\}) \leq v(X_j \setminus \{g\}) \leq v(X_i) = \hat{v}_i(X_i),$$

as desired. □

## 5 Partition Matroids with Different Capacities, Different Binary Valuations

In this section we consider $n$ agents with different binary valuations and partition matroids with different capacity constraints.

**Theorem 5.1.** *For partition matroids with heterogeneous capacities and heterogeneous binary valuations, there exist an F-EF1 allocation for n agents.*

The main tool we use is *priority matchings*, which we define below.

In a graph $G = (V, E)$, a *matching* is a subset $\mu \subseteq E$ such that each vertex $u \in V$ is adjacent to at most a single edge in $\mu$. A vertex adjacent to an edge of $\mu$ is said to be *saturated by* $\mu$. Given a partition of $V$ into subsets $V_1, \ldots, V_k$ (called *priority classes*), a *priority matching* is a matching that, among all feasible matchings, maximizes the number of saturated vertices of $V_1$; subject to this, maximizes the number of saturated vertices of $V_2$; etc. Priority matchings were introduced by Roth, Sönmez, and Ünver (2005) in the context of kidney exchange. They consider the case where each priority class is a singleton. They prove that every priority matching is also a *maximum-cardinality matching*; that is, maximizes the overall number of saturated vertices of $V$. Okumura (2014) extends their results to arbitrary priority classes, and also shows a polynomial-time algorithm for finding a priority matching. Faster algorithms were recently presented by Turner (2015b,a).

The general scheme of our algorithm is presented as Algorithm 4. Each step is explained in detail below.

**Algorithm 4:** F-EF1 allocation with binary valuations.

```
Input: $M, C, k_i^h$ for every $i \in [n], h \in [l]$.
Result: an F-EF1allocation $X$.
Initialize:
$\forall i \in [n]\ X_i \leftarrow \emptyset$
for each category $h$ do
    $\forall i \in [n]\ X_i^h \leftarrow \emptyset$
    Let $T^h := \max_{i \in N} k_i^h$;
    for $t = 1, \ldots, T^h$ do
        Construct a bipartite agent-item graph $G_t^h$
          (see below for details);
        Partition the agents into priority-classes based
          on a topological order on the feasible
          envy-graph;
        Find a priority matching in $G_t^h$;
        For every item $g$ matched to agent $i$ :
          $X_i^h \leftarrow X_i^h \cup \{g\}$;
    end
    Allocate the unmatched items of $C^h$ arbitrarily to
      agents with remaining capacity;
    $\forall i \in [n]\ X_i \leftarrow X_i \cup X_i^h$
end
[TODO: write in a way that is consistent with the
  other algorithms. Maybe add step numbers and refer
  to them in the explanation below.]
```

Algorithm 4 works category-by-category. For each category $h$, the items of $C^h$ are allocated in several iterations, where in each iteration, every agent receives at most one item. The number of required iterations is at most the maximum capacity of an agent in $C^h$; we denote this number by

$T^h := \max_{i \in N} k_i^h$.

In each iteration $t$, we construct a bipartite graph that we denote by $G_t^h$. One side of $G_t^h$ consists of the agents with remaining capacity, i.e., agents for which $k_i^h > |X_i^h|$. [TODO: maybe add the notion of "agent with remaining capacity" to the preliminaries — if it is used elsewhere]. The other side of $G_t^h$ contains the unallocated items of $C^h$. There is an edge between an agent $i$ and an item $j$ iff the agent wants the item, i.e., $v_i(j) = 1$.

We then find a topological order $\sigma$ on the feasible envy-graph (we later prove that the feasible envy-graph never has cycles, so a topological order exists), and partition the vertices of $G_t^h$ corresponding to agents into priority-classes $V_1, \ldots, V_k$ such that, for all $i < j$, the agents in $V_i$ preceed the agents in $V_j$ in $\sigma$.(**Amitay:** I think we need to also make sure there is no envy within each class. otherwise all of the agents could be in the same class) (**Erel:** You are right. In fact, we can make each class a singleton.) In other words, agents in $V_i$ are not envied by agents in $V_j$. This step assumes that $G_t^h$ is cycle-free; we will prove below that it is.

Based on the priority-classes $V_1, \ldots, V_k$, we find a priority matching in $G_t^h$, and allocate each item to the agent matched to it. Then we update the envy-graph and the agent-item graph and proceed to allocate another batch of items of $C^h$.

After at most $T^h$ iterations, no more items of $C^h$ can be allocated to agents who value them at 1, but there may be remaining items that are valued at 0 by all agents with remaining capacity. We allocate these items to arbitrary agents with remaining capacity. This is always possible, since by assumption a feasible allocation exists, so the sum of capacities of all agents is at least $|C^h|$. [TODO: verify this. Do we mention this condition elsewhere?]

To prove the correctness of the algorithm, we need to prove that the envy-graph never has cycles, and that the feasible envy between every two agents is at most 1. We prove both conditions simultaneously.

**Theorem 5.2.** *In each iteration of Algorithm 4:*

*(a) The envy-graph has no cycles;*

*(b) For every $i, j \in N$, the envy of $i$ in $j$ is at most 1. [TODO: Define "envy of A in B" in the preliminaries] (**Amitay:** maybe use PE (positive envy) that we defined in section 6? we can move it to the model section) (**Erel:** Sounds good)*

*Proof.* The proof is by induction. Both claims obviously hold when the algorithm starts. For any category $h$ and iteration $t \in [T^h]$, we denote by *before (after) ht* the situation before (after) iteration $t$ of allocating the items in $C^h$. We assume that properties (a) and (b) hold before $ht$ and prove that they hold after $ht$.

(a) Suppose by contradiction that after $ht$ there is a cycle $i_1 \to \cdots \to i_k = i_1$ in the envy-graph. By assumption (a) the cycle did not exist before $ht$, so at least one edge is due to an item of $C^h$ allocated in iteration $t$. Suppose w.l.o.g. that it is the edge $i_1 \to i_2$. This means that $i_1$ did not receive an item adjacent to it in $G_t^h$, while $i_2$ did receive some item $j_2 \in C^h$ that is adjacent to $i_1$ in $G_t^h$. This $j_2$ was necesssarily adjacent to $i_2$ too in $G_t^h$ — otherwise the priority matching algorithm would assign it to $i_1$ instead. By assumption (b), the envy of $i_2$ in $i_3$ before $ht$ was at most 1, and the utility of $i_2$ increased by 1 due to $j_2$. Therefore, the envy edge $i_2 \to i_3$ implies that $i_3$ received some item $j_3 \in C^h$ that is adjacent to $i_2$ in $G_t^h$. This $j_3$ was necesssarily adjacent to $i_3$ too in $G_t^h$ — otherwise the priority matching algorithm would assign the items $(j_3, j_2)$ to $(i_2, i_1)$ instead. Similar arguments imply that every agent in the cycle must have received an item adjacent to it in $G_t^h$. This includes the agent $i^k = i^1$ — a contradiction.

(b) Suppose by contradiction that the envy of Alice in Bob after $ht$ is more than 1. Since the valuations are binary and at most one item is allocated in each iteration, the envy of Alice in Bob before $ht$ must have been exactly 1. Therefore, Alice appeared before Bob in the topological order $\sigma$. Alice's envy towards Bob increased, so the algorithm must have allocated to Bob some item $j \in C^h$ that was adjacent to Alice in $G_t^h$(**Amitay:** this sentence is not true, alice could become envious more than EF1 even if she is not on the graph because her capacity in this category is full. the contradiction is that if that happened it implies that she was envious more than EF1 even before that category. ). (this implies that Alice both values $j$ at 1 and had sufficient remaining capacity to receive $j$). If Alice did not reach her capacity in category $h$, the algorithm for priority-matching on $G_t^h$ would prefer the matching in which $j$ is given to Alice to the one in which $j$ is given to Bob — a contradiction. Otherwise, since the item allocated to Bob was of value to Alice, it means we couldn't have given Alice any items worth 0 to her in this category, as we would assign this item to her beforehand. So in category $h$, all of the items allocated to Alice are worth 1 to her. That is, if $X$ is the allocation after $ht$, $v_A(X_A^h) = k_A^h \geq \hat{v}_A(X_B^h)$. After $ht$ Alice envies Bob more than EF1, $v_A(X_A) \leq \hat{v}_A(X_B) - 2$. This implies that if $X'$ was the allocation before $h1$, that is- before starting to allocate items from category $h$:

$$\begin{aligned} v_A(X'_A) &= v_A(X_A) - v_A(X_A^h) \\ &\leq \hat{v}_A(X_B) - 2 - \hat{v}_A(X_B^h) \\ &= \hat{v}_A(X'_B) - 2, \end{aligned}$$

implying the allocation was not F-EF1 even before category $h$ — a contradiction.

□

**Remark 5.3.** Consider the special case in which the capacities are binary, i.e., $k_i^h \in \{0, 1\}$ for all $i, h$. Then Algorithm 4 runs a single priority matching in each category. Since this matching is also maximum-cardinality, it maximizes the sum of utilities in each category, and thus maximizes the overall social welfare. Therefore, the algorithm in this case returns a Pareto-efficient allocation. [TODO: check whether we define Pareto-efficient anywhere]

However, when the capacities are larger, the algorithm may return a non Pareto-efficient allocation. For example, suppose there is a single category and two agents with a capacity of 2. There are four items with values $(1, 0), (1, 0), (1, 1), (0, 1)$. In iteration 1, the priority matching may assign $(1, 1)$ to Alice and $(0, 1)$ to Bob. Then Bob does not want any remaining item, so in iteration 2 Alice gets $(0,1)$. The final value profile is $(2,1)$, which is Pareto-dominated by the allocation giving $(1,0),(1,0)$ to Alice and $(1,1),(0,1)$ to Bob. [TODO: put in a table / in the appendix]

# 6 Partition Matroids with Different Capacities, Different Non-Binary Valuations

## 6.1 Different Capacities, Different Valuations, 2 Categories

In this section we present an algorithm that finds an F-EF1 allocation in settings with 2 categories and different capacities and valuations.

**Theorem 6.1.** *For Partition matroids with 2 identical categories, different capacities and different additive valuations there exist a F-EF1 allocation.*[1]

**Algorithm 5:** Bi-directional CRR

```
$\sigma \leftarrow$ an arbitrary order over the agents.
Run Capped Round Robin with $C^1, \sigma$;
Let $X_i^1$ be the outcome allocation for agent $i$;
$\forall i \in [n]\ X_i \leftarrow X_i^1$;
$\sigma' \leftarrow reverse(\sigma)$
Run Capped Round Robin with $C^2, \sigma'$;
Let $X_i^2$ be the outcome allocation for agent $i$;
$\forall i \in [n]\ X_i \leftarrow X_i \cup X_i^2$;
```

*Proof.* Our algorithm (Algorithm 5) runs CRR in an arbitrary order for the first category, then uses the reverse order for CRR in the second category. After the first phase of Algorithm 5 (the first category), by Theorem 4.1, the allocation is F-EF1 and no agent envies another agent that appears in $\sigma$ after her. Consider two arbitrary agents $i,j$ at the end of the algorithm. If agent $i$ f-envied agent $j$ (up to 1 good) after the first category, she appears before $j$ in $\sigma'$ and thus will not gain any more envy in the second category. If she didn't envy after the first category, she can only gain envy up to one good in the second category. That is- in one of the categories she might envy up to one good, in the other she will not envy at all. Adding them together results in a F-EF1 allocation. □

## 6.2 Different Capacities, Different Valuations, 2 agents

In this section we present an algorithm for 2 agents.

**Theorem 6.2** (Different Capacities, Different Valuations, 2 agents)**.** *In every setting with 2 agents, a F-EF1 allocation exists and can be computed efficiently by Algorithm $RR^2$ (Algorithm 6).*

Before we present the algorithm and its analysis, we introduce some notation.

- Given an allocation $X$, the *surplus* of agent $i$ in category $h$ is
$$s_i^h(X) := \hat{v}_i(X_i^h) - \hat{v}_i(X_j^h).$$
I.e., it is the difference between agent $i$'s value for her own bundle and her value for agent $j$'s bundle.
- Given agents $1,2$, $\ell \in \{1,2\}$, valuation functions $v,v'$ and category $h$, $\mathcal{X}(v,v',\ell)^h$ is the allocation obtained by Capped Round Robin (Algorithm **??**) for category $h$, under valuations $v_1 = v, v_2 = v'$, and where agent $\ell$ plays first. When clear in the context, we omit the superscript $h$ from $\mathcal{X}(v,v',\ell)^h$.

We are now ready to present Algorithm "Round Robin Squared" ($RR^2$) . In $RR^2$, there are two layers of round robin (RR), one layer for choosing the next category, and one layer for choosing items within a category. For every agent $i$, the categories are ordered based on $s_i^h(\mathcal{X}(v_1,v_2,i))$, in a non-increasing order; call this order $\pi_i$. In the first iteration, agent 1 chooses the first category in $\pi_1$. Within this category, the items are allocated according to Capped Round Robin (CRR) (Algorithm 2), with agent 1 choosing first. In the second iteration, agent 2 chooses the first category in $\pi_2$ that has not been chosen yet. Within this category, the items are allocated according to CRR, with agent 2 choosing first. The algorithm proceeds in this way, where in every iteration, the agent who chooses the next category flips; that agent chooses the highest category in her order that has not been chosen yet, and within that category, agents are allocated according to CRR with that agent choosing first. This process proceeds until all categories have been exhausted.

**Algorithm 6:** RR-Squared ($(RR)^2$)

```
Input: $a \in \{1,2\}$ the first agent to choose, a set of
       items $M$, categories $C^1,...,C^l$,capacities $k_i^h$
       for every $i = 1,2, h \in [l]$.
$\forall i = 1,2\ X_i \leftarrow \emptyset$;
$\forall i = 1,2\ \psi_i \leftarrow$ Order over the categories according
  to $s_i^h(\mathcal{X}(v_1,v_2,i))$ for every category $h$. (According
  to the expected results of running Algorithm 2 over
  them where $i$ is first to choose).
for $j \in [l]$ do
    $h \leftarrow$ the first category in $\psi_a$ not yet played;
    $X^h \leftarrow \mathcal{X}(v_1,v_2,a)$ (result of CRR (Algorithm 2)
      over category $h$ where $a$ is first to choose);
    $\forall i \in [2]\ X_i \leftarrow X_i \cup X_i^h$;
    Switch $a$ to be the other agent;
end
```

The key lemma in our proof asserts that the surplus of an agent $i$ when playing first within a category $h$ is at least as large as minus the surplus of the same agent when playing second in the same category. I.e.,

**Lemma 6.3.** *For every category $h$ and every $i = 1,2$:*
$$s_i^h(\mathcal{X}(v_1,v_2,i)^h) \geq -s_i^h(\mathcal{X}(v_1,v_2,j)^h)$$

Before proving Lemma 6.3, we show how it implies the assertion of Theorem 6.2.

[1] This can be extended to k categories and F-EF$\frac{k}{2}$, Envy freeness up to $\frac{k}{2}$ goods

*Proof of Theorem 6.2.* We first show that the first agent to choose does not F-envy the other agents. That is,

$$v_i(X_i) \geq \hat{v}_i(X_j),$$

where $i$ is the first agent to choose.

By reordering, let $C^1, ..., C^l$ be the categories in the order they are chosen, and let agent 1 choose a category first. It holds that:

$$v_1(X_1) - \hat{v}_1(X_2) = \sum_{h=1}^{\ell} v_1(X_1^h) - \sum_{h=1}^{\ell} \hat{v}_1(X_2^h)$$

$$= \sum_{h=1}^{\ell}(v_1(X_1^h) - \hat{v}_1(X_2^h))$$

$$= \sum_{h \text{ is odd}} s_1^h(\mathcal{X}(v_1, v_2, 1)) + \sum_{h \text{ is even}} s_1^h(\mathcal{X}(v_1, v_2, 2)) \tag{2}$$

$$\geq \sum_{h \text{ is odd}} s_1^h(\mathcal{X}(v_1, v_2, 1)) + \sum_{h \text{ is even}} -s_1^h(\mathcal{X}(v_1, v_2, 1)) \tag{3}$$

$$= \sum_{t=1}^{\frac{\ell}{2}}(s_1^{2t-1}(\mathcal{X}(v_1, v_2, 1)) - s_1^{2t}(\mathcal{X}(v_1, v_2, 1))). \tag{4}$$

The first equations follow from additivity. Equation 2 follows from the definition of surplus, the facts that agent 1 chooses the odd categories, and the agent who chooses the category is the one to choose first within this category. Inequality 3 follows from Lemma 6.3.

Now, since agent 1 chooses the odd categories, and she chooses according to the highest surplus of the categories not yet chosen, it means that for every $t$ $s_1^{2t-1}(\mathcal{X}(v_1, v_2, 1)) \geq s_1^{2t}(\mathcal{X}(v_1, v_2, 1))$ as category $2t$ was available when agent 1 chose category $2t-1$. Therefore, every summand in the sum of line 4 is non-negative. Thus, the whole sum is non-negative, implying that $v_1(X_1) \geq \hat{v}_1(X_2)$, as desired.

We next show that agent 2 does not F-envy agent 1 beyond F-EF1. As a thought experiment, consider the same setting with the first chosen category removed. Following the same reasoning as above, in this setting agent 2 does not F-envy agent 1. But within the first category, agent 2 can only F-envy agent 1 up to 1 item. That is, there exists one item in the first category such that when it is removed, it eliminates the feasible envy of the second agent within that category, and thus eliminates her total feasible envy. We conclude that the obtained allocation is F-EF1. □

We are now ready to establish the proof of Lemma 6.3. Its proof is based on several lemmas, which we state below.

**(MF:** up to here.)

**Lemma 6.4.** *When two agents with identical additive valuations $v$ play CRR on one category, and one agent plays according to $v$, the best strategy of the other agent is to play according to $v$. that is, for every additive valuation $v'$ and every $\ell \in \{1, 2\}$:*

$$(a) \qquad v(\mathcal{X}(v, v, \ell)_1) \geq v(\mathcal{X}(v', v, \ell)_1)$$

$$(b) \qquad v(\mathcal{X}(v, v, \ell)_2) \geq v(\mathcal{X}(v, v', \ell)_2)$$

*Proof.* The two statements are obviously analogous; below we prove claim (b).

Denote by "truthful play" the play of CRR in which agent 2 plays by $v$ and gets the bundle $\mathcal{X}(v, v, \ell)_2$; denote by "untruthful play" the play of CRR in which agent 2 plays by $v'$ and gets the bundle $\mathcal{X}(v, v', \ell)_2$. Order the items in each of these two bundles in descending order of $v$. Denote the resulting ordered vectors $\pi$ and $\pi'$ respectively, such that $v(\pi_1) \geq v(\pi_2) \geq \cdots$ and $v(\pi'_1) \geq v(\pi'_2) \geq \cdots$. Note that $|\pi| = |\pi'|$ (agent 2 gets the same number of items in both plays). We now prove that $v(\pi_t) \geq v(\pi'_t)$ for all $t \leq |\pi|$.

For every index $t \leq |\pi|$, denote by $z_t$ the number of items held by agent 1 when it's the $t$-th turn of agent 2, that is:

$$z_t := \begin{cases} \min(k_1^h, t-1) & \text{if } \ell = 2 \\ \min(k_1^h, t) & \text{if } \ell = 1 \end{cases}$$

Assume towards contradiction that there exist an index $t \leq |\pi|$ s.t. $v(\pi'_t) > v(\pi_t)$ and let us look at smallest such $t$ (corresponding to a highest valued item in $\pi'$).

In the truthful play, before agent 2 picks $\pi_t$, agents 1 and 2 together hold the $z_t + t - 1$ highest-valued items; hence there are exactly $z_t + t - 1$ items more valuable than $\pi_t$.

In the untruthful play, agent 1 still plays by $v$ and thus still holds at least $z_t$ of the $z_t + t - 1$ highest-valued items. While we do not know by which order agent 2 picks items, we do know that in the final allocation $\pi'$, the first $t$ items are at least as valuable as $\pi'_t$, which is by assumption more valuable than $\pi_t$. Hence there are at least $z_t + t$ items more valuable than $\pi_t$ — a contradiction. □

Since the sum of values of bundle 1 and bundle 2 is fixed, we get the following corollary:

**Lemma 6.5.** *When two agents with identical additive valuation $v$ play CRR on one category, and one agent plays according to $v$, the worst case for this agent is that the other agent plays according to $v$ too. That is, for every $v'$ and $\ell \in \{1, 2\}$:*

$$(a) \qquad v(\mathcal{X}(v, v, \ell)_2) \leq v(\mathcal{X}(v', v, \ell)_2)$$

$$(b) \qquad v(\mathcal{X}(v, v, \ell)_1) \leq v(\mathcal{X}(v, v', \ell)_1)$$

A third lemma that we will need is:

**Lemma 6.6.** *When two agents with identical additive valuation $v_1 = v_2 = v$ play CRR on one category, the value of each agent to her own bundle when she plays first is at least her value to the other agent's bundle when the other agent plays first, and vice versa:*

$$(a) \qquad v_1(\mathcal{X}(v, v, 1)_1) \geq \hat{v}_1(\mathcal{X}(v, v, 2)_2)$$

$$(b) \qquad v_1(\mathcal{X}(v, v, 2)_1) \geq \hat{v}_1(\mathcal{X}(v, v, 1)_2)$$

*Proof.* When both agents play using the same valuation, the only thing that differentiates agent 1's bundle when 1 chooses first/second from agent 2's bundle when 2 chooses first/second is the capacity. If $k_1^h \leq k_2^h$, than $\mathcal{X}(v, v, 1)_1 \subseteq \mathcal{X}(v, v, 2)_2$, and moreover, $\mathcal{X}(v, v, 1)_1 = \textsc{Best}_1(\mathcal{X}(v, v, 2)_2)$ and (a) holds with equality. Otherwise, $\mathcal{X}(v, v, 2)_2 = \textsc{Best}_1(\mathcal{X}(v, v, 2)_2) \subset \mathcal{X}(v, v, 1)_1$ so $v_1(\mathcal{X}(v, v, 1)_1) > \hat{v}_1(\mathcal{X}(v, v, 2)_2)$ and (a) holds strictly. Similar considerations apply to (b). □

Using the same proof from Lemma 6.4, but for agents where both capacities are set to be the minimum of $k_1^h, k_2^h$ proves the following corollary regarding feasible valuations:

**Lemma 6.7.** *When two agents with identical additive valuation $v_1 = v_2 = v$ play $CRR$ on one category, and one agent plays according to $v$, they will feasibly prefer the bundle the other agent gets playing according to $v$ than the one the other agent gets playing according to $v' \neq v$. That is, for every $\ell \in \{1, 2\}$:*

$$(a) \qquad \hat{v}_1(\mathcal{X}(v, v, l)_2) \geq \hat{v}_1(\mathcal{X}(v, v', l)_2)$$

$$(b) \qquad \hat{v}_2(\mathcal{X}(v, v, l)_1) \geq \hat{v}_2(\mathcal{X}(v', v, l)_1)$$

*proof of Lemma 6.3.* We will prove for $i = 1, j = 2$. The other case is analogous. The proof is based on the following four inequalities:

1. $v_1(\mathcal{X}(v_1, v_2, 1)_1) \geq v_1(\mathcal{X}(v_1, v_1, 1)_1)$
2. $\hat{v}_1(\mathcal{X}(v_1, v_2, 1)_2) \leq \hat{v}_1(\mathcal{X}(v_1, v_1, 1)_2)$
3. $v_1(\mathcal{X}(v_1, v_1, 1)_1) \geq \hat{v}_1(\mathcal{X}(v_1, v_2, 2)_2)$
4. $\hat{v}_1(\mathcal{X}(v_1, v_1, 1)_2) \leq v_1(\mathcal{X}(v_1, v_2, 2)_1)$

We prove each inequality separately.

1. Apply Lemma 6.5(b) with $v := v_1$ and $v' := v_2$ and $\ell = 1$.
2. Apply Lemma 6.7(a) with $v := v_1$ and $v' := v_2$ and $\ell = 1$.
3. First, we claim that $v_1(\mathcal{X}(v_1, v_1, 1)_1) \geq \hat{v}_1(\mathcal{X}(v_1, v_1, 2)_2)$. This follows from Lemma 6.6(a) with $v := v_1$.
   Secondly, we claim that $\hat{v}_1(\mathcal{X}(v_1, v_1, 2)_2) \geq \hat{v}_1(\mathcal{X}(v_1, v_2, 2)_2)$. Apply Lemma 6.7(a) with $v := v_1$ and $v' := v_2$ and $\ell = 2$.
4. First, we claim that $\hat{v}_1(\mathcal{X}(v_1, v_1, 1)_2) \leq v_1(\mathcal{X}(v_1, v_1, 2)_1)$. This follows from Lemma 6.6(b) with $v := v_1$.
   Secondly, we claim that $v_1(\mathcal{X}(v_1, v_1, 2)_1) \leq v_1(\mathcal{X}(v_1, v_2, 2)_1)$. Apply Lemma 6.5(b) with $v := v_1$ and $v' := v_2$ and $\ell = 2$.

Combining the 4 inequalities gives:

$$\begin{aligned} s_1^h(\mathcal{X}(v_1, v_2, 1)) &= v_1(\mathcal{X}(v_1, v_2, 1)_1) - \hat{v}_1(\mathcal{X}(v_1, v_2, 1)_2)) \\ &\geq^{1,2} v_1(\mathcal{X}(v_1, v_1, 1)_1) - \hat{v}_1(\mathcal{X}(v_1, v_1, 1)_2) \\ &\geq^{3,4} \hat{v}_1(\mathcal{X}(v_1, v_2, 2)_2) - v_1(\mathcal{X}(v_1, v_2, 2)_1) \\ &= -s_1^h(\mathcal{X}(v_1, v_2, 2)), \end{aligned}$$

which completes the proof of the lemma. □

## 7 General Matroids with up to 3 agents

In this section we consider general matroid constraints with identical matroids and heterogeneous valuations. We first observe that every instance with 2 agents admits an F-EF1 allocation.

**Observation 7.1.** *For identical general matroid constraints, 2 agents and heterogeneous valuations there exists an F-EF1 allocation.*

*Proof.* Observe the case of 2 agents with valuations $v_1, v_2$. (Biswas and Barman 2019) devise a method for EF1 allocation for identical matroids and identical valuations. Use this method on an instance where both agents have an identical valuation $v_1$, to partition the items into two bundles. Both bundles are feasible for both agents as they share identical matroid constraints. Let agent 2 choose which of the bundles to get, so she cannot be envious at all. Agent 1 gets the other bundle, but she cannot envy more than EF1 as if she was it would mean the allocation is not EF1 even for identical valuations. □

In what follows, we establish an existence result for 3 agents with binary valuations.

**Theorem 7.2.** *For identical general matroid constraints and 3 agents with heterogeneous binary valuations, there exists an F-EF1 allocation. Furthermore, there exist such allocation that is also social welfare maximizing.*

Let $X$ be a welfare-maximizing allocation. We construct $X'$ which is an EF1 allocation that is also welfare maximizing. The algorithm we use is as follows: while there exist agents $i, j$ such that $i$ envies $j$ beyond EF1, if possible move an item valuable to $i$ from $j$ to $i$. otherwise- swap items such that $i$ will get an item worth 1 to her and give an item worth 0 to her. We prove that one of the options is always possible and that the process terminates, but first we need several lemmas.

**Algorithm 7:** Repeated Swaps

```
Initialize:
    $X \leftarrow$ a welfare maximizing complete allocation;
while $X$ is not EF1 do
    Find $i, j$ s.t. $Envy^+(i, j) > 1$;
    If possible, move item $a$ s.t. $v_i(a) = 1$ from $X_j$
      to $X_i$;
    Else, swap between items $a \in X_j, b \in X_i$ s.t.
      $v_i(a) = 1, v_i(b) = 0$
end
```

**Definition 7.3.** Given a matroid $(M, \mathcal{I})$ and 2 independent sets $I, J \in \mathcal{I}$, items $i \in I$ and $j \in J$ represent a *feasible swap* if both $(J \setminus \{j\}) \cup \{i\}$ and $(I \setminus \{i\}) \cup \{j\}$ are in $\mathcal{I}$.

**Lemma 7.4** (Biswas et al.)**.** *Let $(M, \mathcal{I})$ be a matroid with independent subsets $I, J \in \mathcal{I}$ which satisfy $I \cap J = \emptyset$ and $|I| \geq |J|$. Then, there exists a one-to-one map $\mu : J \to I$ such that for any $j \in J$, the pair $j, \mu(j)$ represents a feasible swap.*

**Lemma 7.5.** *For a matroid constrained setting with binary valuations, if agent $i$ envies agent $j$ under a feasible allocation $X$, one of these 2 options hold:*
*1) There exist $a \in X_j$ s.t. $v_i(a) = 1$ and $a$ can be added to $i$'s bundle: $X_i \cup \{a\} \in \mathcal{I}$*
*2) There exists $a \in X_j$, $b \in X_i$ s.t. $v_i(a) = 1$, $v_i(b) = 0$ and $a, b$ represent a feasible swap.*
*That is, there exist either an item that can be feasibly moved from $j$ to $i$ or two items that can be feasibly exchanged between $i$ and $j$ such that agent $i$'s value will increase by 1*

*from the action. We will denote such item swap/transfer a* smart swap.

*Proof.* Case 1: $|X_i| \geq |X_j|$; By Lemma 7.4, there exist a matching between $X_j$ and some subset of $X_i$ such that any matched pair is a feasible swap. Agent $i$ is envious of agent $j$, that means

$$|\{a \in X_j : v_i(a) = 1\}| = v_i(X_j) >$$
$$> v_i(X_i) = |\{a \in X_i : v_i(a) = 1\}|,$$

By the pigeonhole principle, in any such matching there will be at least one pair of matched items $a \in X_j$, $b \in X_i$ s.t. $v_i(a) = 1$, $v_i(b) = 0$. Consequently, property (2) of the lemma holds.

Case 2: $|X_i| < |X_j|$; By the matroid exchange property, there exists $c \in X_j$ s.t. $X_i \cup \{c\} \in \mathcal{I}$. If there exists such $c$ s.t. $v_i(c) = 1$, property (1) holds. Otherwise, for all such $c : v_i(c) = 0$ and let us denote the group of those items by $C = \{c \in X_j \text{ s.t. } X_i \cup \{c\} \in \mathcal{I}\}$. Consider some group $D \subseteq X_j$ such that $X_i \cup D \in \mathcal{I}$, and $|X_i \cup D| = |X_j|$. Such a group must exist from the matroid exchange property. We now notice that $D \subseteq C$, as any subgroup of $X_i \cup D$ is also an independent set due to matroid hereditary property, in particular for every $d \in D$ $X_i \cup \{d\} \in \mathcal{I}$, so every such $d$ is also in $C$. Hence $v_i(D) = 0$.
Now consider the independent sets $X_j, (X_i \cup D)$. It is shown in (Goemans 2009) that for any two independent subsets of the same cardinality, $I$ and $J$, there exists a bijection $\mu : J \setminus I \to I \setminus J$ such that any matched pair represent a feasible swap.
In our case, if we take $J = X_j, I = X_i \cup D$ the bijection is between $J \setminus I = X_j \setminus D$ and $I \setminus J = X_i$. Since $v_i(X_j \setminus D) = v_i(X_j) - v_i(D) = v_i(X_j) > v_i(X_i)$, a similar argument to case 1 suggests in such bijection there must be some item $a \in X_j$ such that $v_i(\mu(a)) = 0, v_i(a) = 1$. Consequently, property (2) of the lemma holds.

□

**Lemma 7.6.** *Let $X$ be a* welfare maximizing allocation *in a matroid constrained setting with binary valuations, and let agent $i$ envy agent $j$ under that allocation. Lemma 7.5 holds with the additional properties:*
*1) $v_j(a) = 1$.*
*2) if the second option of Lemma 7.5 holds, $v_j(b) = 0$.*
*i.e. one can either move item $a$ that is worth $1$ to both agents from $X_j$ to $X_i$ or swap between item $a$ and some item $b \in X_i$ that is worth $0$ to both agents.*

*Proof.* From Lemma 7.5 we know that there must be a feasible smart swap. Denote the item moved from $j$ to $i$: $a$, denote the item that moved from $i$ to $j$ (if a swap was needed) $b$.
By Lemma 7.5, $v_i(a) = 1$ and $v_i(b) = 0$. Regarding $v_j$, notice that if $v_j(a) = 0$ or $v_j(b) = 1$ the swap would result in a strictly higher social welfare, in contradiction to the maximality of $X$. Hence $v_j(a) = v_i(a) = 1$ and $v_j(b) = v_i(b) = 0$. □

**Corollary 7.7.** *For a welfare maximizing allocation $X$ where agent $i$ envies agent $j$, every smart swap between $j$ and $i$ holds that the resulting allocation $X'$ will remain welfare maximizing.*

$$SW(X') = SW(X) - v_i(b) + v_i(a) - v_j(a) + v_j(b)$$
$$= SW(X) - 0 + 1 - 1 + 0 = SW(X)$$

Therefore, at every step the allocation remains welfare maximizing and we can re-apply Lemma 7.6 to make another smart swap.

**Lemma 7.8.** *Let $X$ be a welfare maximizing allocation where agent $i$ envies agent $j$ such that $Envy^+(i,j) > 1$, and let $X'$ be the allocation after a smart swap.*

1. *$Envy^+(i,j)$ will drop by 2 after the smart swap.*
2. *$Envy^+(j,i)$ after the smart swap is 0.*

*Proof.* 1. Let $a$ be the item added to $X_i$ in the smart swap.

$$v_i(X_j') = v_i(X_j) - v_i(a) = v_i(X_j) - 1,$$
$$v_i(X_i') = v_i(X_i) + v_i(a) = v_i(X_i) + 1.$$

If we denote $Envy^+(i,j)'$ the positive envy after the smart swap:

$$Envy^+(i,j)' = v_i(X_j') - v_i(X_i')$$
$$= v_i(X_j) - 1 - (v_i(X_i) + 1) = Envy^+(i,j) - 2$$

2. Notice that $v_i(X_j') - v_i(X_i') \geq 0$ after the smart swap. If $j$ would start to envy $i$, switching their bundles would strictly increase social welfare in contradiction to maximality.

□

**(Erel:** I suggest to write the algorithm in a float.)
Now we prove theorem 7.2

*Proof.* Let $X$ be a complete feasible welfare-maximizing allocation.

If $X$ is EF1, we are done. Otherwise, at each iteration of the algorithm, choose agent $i$ that envies some agent $j$ such that $Envy^+(i,j) > 1$. If $j$ also envies $i$ we could switch their bundles and receive higher SW in contradiction to the assumption, so we can assume $j$ doesn't envy $i$.

Define a potential function:

$$\Phi(X) := \sum_i \sum_{j \neq i} Envy^+(i,j).$$

Intuitively, $\Phi$ is the sum of positive envy in the allocation. By Corollary 7.7 and Lemma 7.8, there must exist a feasible smart swap between $j, i$ such that the social welfare remains unchanged, $Envy^+(i,j)$ drops by 2 and $Envy^+(j,i)$ remains 0. Thus the sum of the relevant terms in the potential, namely $Envy^+(i,j) + Envy^+(j,i)$, drops by 2.

Now let us look at the positive envy that might be added due to other terms of $\Phi$. Denoting the $3^{rd}$ agent as $k$:

1. $Envy^+(i,k)$ cannot increase, as the smart swap improves $i$'s valuation, and does not change $k$'s bundle.
2. $Envy^+(k,i)$ may increase by at most 1, as the largest possible increase in $v_k(X_i')$ is 1, while $v_k(X_k)$ doesn't change.

3. $Envy^+(k, j)$ may increase by at most 1, as the largest possible increase in $v_k(X'_j)$ is 1, while $v_k(X_k)$ doesn't change.
4. $Envy^+(j, k)$ may increase by at most 1, as this is the exact decrease in $v_j(X'_j)$, while $v_j(X_k)$ doesn't change.

Now, notice that among the only options of added envy (#2, #3, #4) no two options can occur simultaneously:

- $Envy^+(k, j), Envy^+(j, k)$ cannot increase simultaneously as this will create an envy cycle, and de-cycling it will strictly increase social welfare in contradiction to maximality.
- $Envy^+(k, i), Envy^+(j, k)$ cannot increase simultaneously as this together with the fact that the envy of $i$ towards $j$ after the smart swap is non-negative will create a weak envy cycle, and de-cycling it will strictly increase social welfare in contradiction to maximality.
- $Envy^+(k, i), Envy^+(k, j)$ cannot increase simultaneously as $v_k(X_i)+v_k(X_j) = v_k(X'_i)+v_k(X'_j)$. That is, in the eyes of agent $k$ the added value to one agent's bundle will strictly decrease the value of the other's bundle.

By the above observation, at every iteration the potential function drops by at least 1: $Envy^+(i, j)$ drops by 2, $Envy^+(j, i), Envy^+(i, k)$ do not change, and among $Envy^+(k, i), Envy^+(k, j), Envy^+(j, k)$ only one can increase, by at most 1. Noticing that $\Phi$ is bounded by 0, the process ends and we end up with a feasible allocation that is EF1 and welfare maximizing. □

## References

Aleksandrov, M.; Aziz, H.; Gaspers, S.; and Walsh, T. 2015. Online fair division: analysing a food bank problem. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 2540–2546.

Amanatidis, G.; Birmpas, G.; Filos-Ratsikas, A.; Hollender, A.; and Voudouris, A. A. 2020. Maximum Nash welfare and other stories about EFX. *arXiv preprint arXiv:2001.09838* .

Babaioff, M.; Ezra, T.; and Feige, U. 2020. Fair and Truthful Mechanisms for Dichotomous Valuations. *arXiv preprint arXiv:2002.10704* .

Barman, S.; Biswas, A.; Krishnamurthy, S. K.; and Narahari, Y. 2017. Groupwise maximin fair allocation of indivisible goods. *arXiv preprint arXiv:1711.07621* .

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 7–13.

Barrera, R.; Nyman, K.; Ruiz, A.; Su, F. E.; and Zhang, Y. X. 2015. Discrete envy-free division of necklaces and maps. ArXiv preprint 1510.02132.

Bei, X.; Igarashi, A.; Lu, X.; and Suksompong, W. 2019. The Price of Connectivity in Fair Division. ArXiv preprint 1908.05433.

Benabbou, N.; Chakraborty, M.; Igarashi, A.; and Zick, Y. 2020. Finding Fair and Efficient Allocations When Valuations Don't Add Up. *arXiv preprint arXiv:2003.07060* .

Bilò, V.; Caragiannis, I.; Flammini, M.; Igarashi, A.; Monaco, G.; Peters, D.; Vinci, C.; and Zwicker, W. S. 2018. Almost Envy-Free Allocations with Connected Bundles. In Blum, A., ed., *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 14:1–14:21. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-095-8. ISSN 1868-8969. doi:10.4230/LIPIcs.ITCS.2019.14. URL http://drops.dagstuhl.de/opus/volltexte/2018/10107.

Biswas, A.; and Barman, S. 2018. Fair Division Under Cardinality Constraints. In *IJCAI*, 91–97.

Biswas, A.; and Barman, S. 2019. Matroid Constrained Fair Allocation Problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9921–9922.

Bouveret, S.; Cechlárová, K.; Elkind, E.; Igarashi, A.; and Peters, D. 2017. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 135–141.

Bouveret, S.; and Lemaître, M. 2016. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems* 30(2): 259–290.

Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D. 2016. *Handbook of computational social choice*. Cambridge University Press.

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

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)* 7(3): 1–32.

Darmann, A.; and Schauer, J. 2015. Maximizing Nash product social welfare in allocating indivisible goods. *European Journal of Operational Research* 247(2): 548–559.

Endriss, U. 2017. *Trends in computational social choice*. Lulu.com.

Ferraioli, D.; Gourvès, L.; and Monnot, J. 2014. On regular and approximately fair allocations of indivisible goods. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 997–1004.

Garg, N.; Kavitha, T.; Kumar, A.; Mehlhorn, K.; and Mestre, J. 2010. Assigning papers to referees. *Algorithmica* 58(1): 119–136.

Goemans, M. 2009. Lecture notes on Matroid Intersection (Lecture 11). *Massachusetts Institute of Technology, Combinatorial Optimization* 13.

Goldman, J.; and Procaccia, A. D. 2015. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges* 13(2): 41–46.

Gourvès, L.; and Monnot, J. 2019. On maximin share allocations in matroids. *Theoretical Computer Science* 754: 50–64. Preliminary version appeared in CIAC 2017.

Gourvès, L.; Monnot, J.; and Tlilane, L. 2013. A protocol for cutting matroids like cakes. In *International Conference on Web and Internet Economics*, 216–229. Springer.

Gourvès, L.; Monnot, J.; and Tlilane, L. 2014. Near Fairness in Matroids. In *ECAI*, 393–398.

Halpern, D.; Procaccia, A. D.; Psomas, A.; and Shah, N. 2020. Fair division with binary valuations: One rule to rule them all. *arXiv preprint arXiv:2007.06073* .

Klaus, B.; Manlove, D. F.; and Rossi, F. 2016. *Matching under preferences*. Cambridge University Press.

Lian, J. W.; Mattei, N.; Noble, R.; and Walsh, T. 2017. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. *arXiv preprint arXiv:1705.06840* .

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM conference on Electronic commerce*, 125–131.

Long, C.; Wong, R. C.-W.; Peng, Y.; and Ye, L. 2013. On good and fair paper-reviewer assignment. In *2013 IEEE 13th International Conference on Data Mining*, 1145–1150. IEEE.

Mackin, E.; and Xia, L. 2016. Allocating indivisible items in categorized domains. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 359–365.

Nyman, K.; Su, F. E.; and Zerbib, S. 2020. Fair division with multiple pieces. *Discrete Applied Mathematics* 283: 115–122.

Okumura, Y. 2014. Priority matchings revisited. *Games and Economic Behavior* 88: 242–249.

Roth, A. E.; Sönmez, T.; and Ünver, M. U. 2005. Pairwise kidney exchange. *Journal of Economic theory* 125(2): 151–188.

Sikdar, S.; Adali, S.; and Xia, L. 2017. Mechanism Design for Multi-Type Housing Markets. In *AAAI*, 684–690.

Sikdar, S.; Adalı, S.; and Xia, L. 2019. Mechanism Design for Multi-Type Housing Markets with Acceptable Bundles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2165–2172.

Suksompong, W. 2019. Fairly allocating contiguous blocks of indivisible items. *Discrete Applied Mathematics* 260: 227–236.

Turner, J. 2015a. Faster Maximium Priority Matchings in Bipartite Graphs. *arXiv preprint arXiv:1512.09349* .

Turner, J. 2015b. Maximium Priority Matchings. *arXiv preprint arXiv:1512.08555* .

# APPENDIX