Journal of Artificial Intelligence Research 76 (2023) 567–611 Submitted 03/2022; published 02/2023

On Fair Division under Heterogeneous Matroid Constraints

Amitay Dror AMITAYDR@GMAIL.COM
Michal Feldman MFELDMAN@TAU.AC.IL
Tel-Aviv University, Tel-Aviv, Israel

Erel Segal-Halevi ERELSGL@GMAIL.COM
Ariel University, Ariel 40700, Israel

Abstract

We study fair allocation of indivisible goods among additive agents with feasibility constraints. In these settings, every agent is restricted to get a bundle among a specified set of feasible bundles. Such scenarios have been of great interest to the AI community due to their applicability to real-world problems. Following some impossibility results, we restrict attention to matroid feasibility constraints that capture natural scenarios, such as the allocation of shifts to medical doctors and the allocation of conference papers to referees.

We focus on the common fairness notion of envy-freeness up to one good (EF1). Previous algorithms for finding EF1 allocations are either restricted to agents with identical feasibility constraints or allow free disposal of items. An open problem is the existence of EF1 complete allocations among agents who differ both in their valuations and in their feasibility constraints. In this work, we make progress on this problem by providing positive and negative results for several matroid and valuation types. Among other results, we devise polynomial-time algorithms for finding EF1 allocations in the following settings: (i) n agents with heterogeneous (non-identical) binary valuations and partition matroids with heterogeneous capacities; (ii) two agents with heterogeneous additive valuations and partition matroids with heterogeneous capacities; and (iii) three agents with heterogeneous binary valuations and identical base-orderable matroid constraints.

1. Introduction

Many real-life problems involve the fair allocation of indivisible items among agents with different preferences, and with constraints on the bundle that each agent may receive. Examples include the allocation of course seats among students (Budish, Cachon, Kessler, & Othman, 2017) and the allocation of conference papers among referees (Garg, Kavitha, Kumar, Mehlhorn, & Mestre, 2010).

In general, the constraints may be *heterogeneous*, that is, different agents may have different constraints. For example, consider the allocation of employees among departments of a company: one department has room for four project managers and two backend engineers, while another department may have room for three backend engineers and five data scientists. Another example can be found in the way shifts are assigned among medical doctors, where every doctor has her own schedule limitations. A third example is allocating course seats among students, where the students' constraints for course categories depend on their curricular year and subject of study.

Our goal is to devise algorithms that find fair allocations of indivisible items among agents with heterogeneous preferences and feasibility constraints. Let us first explain what we mean by "fair" and what we mean by "constraints".

A classic notion of fairness is *envy-freeness* (EF), which means that every agent (weakly) prefers his or her bundle to that of any other agent. Since an EF allocation may not exist when items are indivisible, recent studies focus on its relaxation known as *EF1* — envy-free up to one item (Budish, 2011) — which means that 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 (see Section 2.2 for a formal definition). Without constraints, an EF1 allocation always exists and can be computed efficiently (Lipton, Markakis, Mossel, & Saberi, 2004).

The constraints of an agent are represented by a set of bundles (subsets of items), that are considered *feasible* for the agent. An allocation is feasible if it allocates to each agent a feasible bundle. We focus on the case when the feasible bundles are the *independent sets of a matroid*. This means that (i) the set of feasible bundles is *downward-closed* — a subset of a feasible bundle is feasible; (ii) if a feasible bundle S has fewer items than another feasible bundle T, then it is possible to extend S to a larger feasible bundle by adding some item from T. This latter property of "extension by one" makes the notion of EF1 particularly appropriate for problems of allocation with matroid constraints. A special case of a matroid is a *partition matroid*. With partition matroid constraints, the items are partitioned into a set of *categories*, each category has a *capacity*, and the feasible bundles are the bundles in which the number of items from each category is at most the category capacity.

There are two approaches for handling feasibility constraints in fair allocation. A first approach is to directly construct allocations that satisfy the constraints, i.e., guarantee that each agent receives a feasible bundle. This approach was taken recently by Biswas and Barman (2018, 2019), who study settings with *additive valuations*, where every agent values each bundle at the sum of the values of its items. They present efficient algorithms for computing EF1 allocations when agents have: (i) identical matroid constraints and identical valuations; or (ii) identical partition matroid constraints, even under heterogeneous (i.e., non-identical) valuations (see Section 2.3 for a more detailed description of their algorithm). However, their algorithms do not handle heterogeneous partition constraints, and do not handle identical matroid constraints with heterogeneous valuations.

A second approach is to capture the constraints within the valuation function. That is, the value of an agent for a bundle equals the value of the best feasible subset of that bundle. This approach seamlessly addresses heterogeneity in both constraints and valuations. The valuation functions constructed this way are no longer additive, but are *submodular* (Oxley, 2006). Recently, Babaioff, Ezra, and Feige (2021) and Benabbou, Chakraborty, Igarashi, and Zick (2021) have independently proved the existence of EF1 allocations in the special case in which agents have submodular valuations with *binary marginals* (where adding an item to a bundle adds either 0 or 1 to its value). Such an allocation can be converted to a fair *and feasible* allocation by giving each agent the best feasible subset of his/her allocated bundle, and disposing of the other items.

However, in some settings, such disposal of items may be impossible. For example, when allocating shifts to medical doctors, if an allocation rule returns an infeasible allocation and shifts are disposed to make it feasible, the emergency-room might remain understaffed. A similar problem may occur when allocating papers to referees, where disposals may leave some papers without reviews. The allocation rules developed in the above papers may not yield EF1 allocations when they are constrained to return feasible allocations (Section 8.1). Thus, an open problem remains:

***Open problem**: Given agents with* heterogeneous *additive valuations and* heterogeneous *matroid constraints, which settings admit a* complete *and* feasible *EF1 allocation?*

1.1 Contribution and Techniques

Feasible-envy. Before presenting our results, we shall discuss what it means to "envy" in settings with heterogeneous constraints.

As an example, consider an allocation of 8 seats in programming courses among two students: Alice — an industrial engineering student who is allowed to take only 3 courses; and Bob — a computer science student who is allowed to take 5 courses. The only feasible allocation is giving 3 courses to Alice and 5 courses to Bob. Does Alice envy Bob?

- On one hand, Alice might envy Bob even if all courses have the same value. Moreover, Alice might envy the very fact that she is allowed to take fewer courses than Bob, that is, she might envy Bob's constraints. This kind of envy cannot be avoided without violating the university rules, which is beyond the scope of this paper.

- On the other hand, one may claim that envy is only justified between people who are a-priori equal. Since Alice is subject to different constraints than Bob, she should not envy him regardless of which courses she receives, and therefore we should not aim to avoid such envy.

We take a middle ground between these two extremes: Alice is happy with the constraints on IE students (after all, she chose to study IE and not CS), but within these constraints, she would like to receive high-value courses. Therefore, if her three courses are as valuable as Bob's best three courses, then she does not envy; but if she receives three low-value courses while Bob receives five high-value courses, she is envious.[1]

In general, we say that Alice has a *feasible-envy* towards Bob if she prefers some subset of Bob's bundle, which is feasible for her, over her own bundle. An allocation is *feasible-envy-free (F-EF)* if no agent has a feasible-envy towards another agent; it is *feasible-envy-free up to one item (F-EF1)* if it becomes F-EF after removing at most one item from an envied agent bundle (see Section 2.2 for a formal definition). Note that F-EF1 is equivalent to EF1 when agents have identical constraints.

Throughout the paper, we use the notion of F-EF1 under heterogeneous constraints. In Appendix A we present some alternative (weaker) notions of feasible-envy-freeness, and show that all our results are valid for these weaker definitions too.

Impossibilities. We present several impossibility results that direct us to the interesting domain of study. First, 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 (Section 8.2). Second, going beyond matroid constraints to natural generalizations such as

1. As a more figurative example, consider a restaurant with two people having different eating capacities. The low-capacity person would not envy the fact that the high-capacity person eats more food than he could digest, but he may envy the fact that the high-capacity person eats the tastier food.

matroid intersection, *bipartite graph matching*, *conflict graph* or *budget constraints* is futile: even with two agents with identical binary valuations and identical non-matroid constraints, a complete F-EF1 allocation may not exist (Section 8.3). Third, going beyond EF1 to the stronger notion of *envy-free up to any good (EFX)* is also hopeless: even with two agents with identical valuations and identical uniform matroid constraints, an EFX allocation may not exist (Section 8.4).

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; or (2) *base-orderable (BO) matroids* — a wide class of matroids containing partition matroids — where all agents have identical matroid constraints but possibly different valuations.

Algorithms (see Table 1). For *partition matroids*, the reason that the algorithms of Lipton et al. (2004) and Biswas and Barman (2018) fail for agents with heterogeneous capacities is that they rely on *cycle removal* in the envy-graph. Informally (Section 2.3 for details), these algorithms maintain a directed *envy-graph* in which each agent points to every agent he or she 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, the envy-graph must contain a cycle, which is then removed by exchanging bundles along the cycle. However, when agents in the cycle have different constraints, this exchange may not be feasible. Thus, our main challenge is to develop techniques that guarantee that no envy-cycle are created in the first place. We manage to do so in four settings of interest:

1. There are at most *two categories* (Section 3).
2. All agents have *identical* valuations (Section 4).
3. All agents have *binary valuations* (Section 5).
4. There are *two agents* (Section 6).

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

Beyond partition matroids, we consider the much wider class of matroids, termed *base-orderable* (BO) matroids (see Definition 15). This class contains partition matroids, laminar matroids (an extension of partition matroids where the items in each category can be partitioned into sub-categories), transversal matroids and other interesting matroid structures. For this class we present algorithms for agents with identical constraints and heterogeneous additive valuations in the following cases:

5. There are *three agents* with *binary valuations* (Section 7.3).
6. There are *two agents* (Section 7.4).

We also present an algorithm for agents with *unary valuations* (all items are worth 1 for all agents) and heterogeneous constraints based on general matroids (Section 7.5).

Our algorithms for partition matroids are strongly-polynomial — they use a polynomial number of arithmetic operations and polynomial space. The algorithms for general matroids require, in addition to a polynomial number of arithmetic operations, a polynomial number

Matroid type	Complete allocation	Het. constraints	Het. valuations	PE	valuations	# of agents	Remarks	Reference
Partition	✓	-	✓	-	General	n		*** B&B ⊕
	✓	-	✓	✓	General	2	Pos. & Neg. valuations	*** SSH ⊕
	✓	✓	✓	-	General	n	≤ 2 categories	Section 3, Theorem 3 ⊕
	✓	✓	-	✓	General	n		Section 4, Theorem 4 ⊕
	✓	✓	✓	✓Binary capacities	Binary	n		Section 5, Theorem 6 ⊕
	✓	✓	✓	-	General	2		Section 6, Theorem 8 ⊕
	✓	✓Het. categories	Even identical	-	Even binary	2		No F-EF1; Example 4 ⊖
	✓	Even identical	Even identical	-	Even binary	2		No EFX; Example 8 ⊖
Beyond Partition	✓	-	-	-	General	n	Laminar matroid	*** B&B ⊕
	-	✓	✓	✓	Binary	n	General matroid	*** BCIZ, BEF ⊕
	✓	✓	✓	✓	Unary	n	General matroid	Section 7.5, Theorem 12 ⊕
	✓	-	✓	✓	Binary	3	BO matroid	Section 7.3, Theorem 10 ⊕
	✓	-	-	-	General	n	BO matroid	Section 7.4, Theorem 11 ⊕
	✓	-	✓	-	General	2	BO matroid	Section 7.4 ⊕
	✓	Even identical	Even identical	-	Even binary	2	Matroid intersection; Conflict graph; Budget	No F-EF1; Examples 6, 7 ⊖

Table 1: A summary of our results in the context of previous results. All results are for additive valuations. PE refers to outcomes that are also Pareto-efficient. BO refers to base-orderable matroids (Section 7). Asterisks *** denote results by other authors: B&B is Biswas and Barman (2018), BCIZ,BEF is Benabbou et al. (2021), Babaioff et al. (2021) (see Section 1), and SSH is Shoshan et al. (2023). ⊕ denotes a positive result; ⊖ denotes a negative result.

of calls to an *independence oracle* — a function that checks whether a given subset is an independent set of the matroid.

1.2 Related Work

A recent survey of constraints in fair division is given by Suksompong (2021). Below we focus on constraints in allocation of indivisible items.

Capacity constraints. In many settings, there are lower bounds as well as upper bounds on the total number of items allocated to an agent. This is particularly relevant to the problem of *assigning conference papers to referees* (Garg et al., 2010; Long, Wong, Peng, & Ye, 2013; Lian, Mattei, Noble, & Walsh, 2018). The constraints may be different for each agent, but there is only one category of items. The same is true in the setting studied by Ferraioli, Gourvès, and Monnot (2014), where each agent must receive exactly k items. A *balanced allocation* is an allocation in which all agents receive the same number of items up to at most a single item. The Round Robin algorithm finds a balanced EF1 allocation for any number of agents with additive valuations. An algorithm by Kyropoulou, Suksompong, and Voudouris (2020) finds a balanced EF1 allocation for two agents with general monotone valuations. It is open whether this result extends to three or more agents. Jojic, Panina, and Zivaljevic (2021) prove the existence of a balanced allocation in which all agents assign approximately the same value to all bundles (an "almost consensus" allocation). Note that constraints imposing a lower bound on the number of items allocated to an agent, such as balancedness constraints, are not matroid constraints, since they are not downward-closed.

Capacity constraints are common also in matching markets such as doctors–hospitals and workers–firms; see Klaus, Manlove, and Rossi (2016) for a recent survey. 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.

Gafni, Huang, Lavi, and Talgam-Cohen (2021) study fair allocation in a related setting in which items may have multiple copies.

Partition constraints. Fair allocation of items of different categories is studied by Mackin and Xia (2016) and Sikdar, Adalı, and Xia (2017). Each category contains 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. The above works focus on designing strategyproof mechanisms. 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").

Following the paper by Biswas and Barman (2018) focusing on EF1 fairness, Hummel and Hetland (2022b) study partition matroid constraints in combination with a different fairness notion — the *maximin-share* (Budish, 2011); informally, the maximin-share of an agent is the maximum over all partitions, of the minimum value of a bundle in that partition. Their algorithm attains a 1/2-factor approximation to this fairness notion for general partition matroids, and a 2/3-factor approximation for the special case of partition matroids with a single category.

In a recent paper, Shoshan et al. (2023) presented an algorithm for finding a complete Pareto-efficient and EF1 allocation under identical partition-matroid constraints, when there are two agents with additive valuations that can be positive and negative.

Matroid constraints. Gourvès, Monnot, and Tlilane (2013) study a setting with a single matroid, where the goal includes building a base of the matroid and providing 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 by design requires to leave some items unallocated, which is not allowed in our setting.

Budget constraints. *Budget constraints* (also called *knapsack constraints*) assume that each item has a cost, each agent has a budget, and the feasible bundles for an agent are the bundles with total cost at most the budget. In this setting, Wu, Li, and Gan (2021) show that a 1/4-factor EF1 allocation exists. Gan, Li, and Wu (2021) show that a 1/2-factor EF1 allocation exists when the valuations are identical, and an EF1 allocation exists among two agents with the same budget.

Connectivity constraints. Barrera, Nyman, Ruiz, Su, and Zhang (2015), Bilò, Caragiannis, Flammini, Igarashi, Monaco, Peters, Vinci, and Zwicker (2018), and Suksompong (2019) study another kind of constraint in fair allocation. The goods are arranged on a line, and each agent must receive a connected subset of the line. Bouveret, Cechlárová, Elkind, Igarashi, and Peters (2017) and Bei, Igarashi, Lu, and Suksompong (2022) 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.

No-conflict constraints. Li, Li, and Zhang (2021) study fair allocation with *scheduling constraints*, where each item is an interval, and the feasible bundles are the sets of non-overlapping intervals. Hummel and Hetland (2022a) study fair allocation in a more general setting in which there is a *conflict graph* G, and the feasible sets are the sets of non-adjacent vertices in G.

Downward-closed constraints. Li and Vetta (2021) study fair allocation with *downward-closed constraints*, which include matroid, budget, and no-conflict constraints as special cases. For this very general setting, they present an algorithm that approximates the maximin share.

Non-additive valuations. As explained in the introduction, fair allocation with constraints is closely related (though not equivalent) to *fair allocation with non-additive valuations*. This problem has attracted considerable attention recently. Bei, Garg, Hoefer, and Mehlhorn (2017) and Anari, Mai, Gharan, and Vazirani (2018) study allocation of multi-unit item-types when the valuation is additive between types but concave (i.e., has decreasing marginal returns) between units of the same type. They give a 2-approximation to the maximum Nash welfare (the product of utilities). Garg, Hoefer, and Mehlhorn (2018) study budget-additive valuations, where each agent has a utility-cap and values each bundle as the minimum between the sum of item values and the utility-cap. They give a 2.404-approximation to the maximum Nash welfare.

Particularly relevant to matroid constraints are the *submodular valuations with binary marginals*, where adding an item to a bundle increases the bundle value by either 0 or 1. These valuations are equivalent to *matroid rank functions* — functions that evaluate a bundle by the size of the largest independent set of a certain matroid contained in that bundle. In this setting, Benabbou et al. (2021) and Babaioff et al. (2021) present allocation mechanisms that are Pareto-efficient, EF1, EFX and strategyproof. Barman and Verma (2021) present a polynomial-time algorithm that finds an allocation satisfying maximin-share fairness.

2. Model and Preliminaries

2.1 Allocations and Constraints

We consider settings where a set M of m items should be allocated among a set N of n agents. An allocation is denoted by $\mathbf{X} = (X_i)_{i\in N}$ where $X_i \subseteq M$ is the bundle given to agent i, and $X_i \cap X_j = \emptyset$ for all $i \neq j \in N$. An allocation is *complete* if $\biguplus_{i\in N} X_i = M$. Throughout, we use $[k]$ to denote the set $\{1, \dots, k\}$ for every positive integer k.

We consider constrained settings where every agent i is associated with a matroid $\mathcal{M}_i = (M, \mathcal{I}_i)$ that specifies the feasible bundles for i.

Definition 1. A *matroid* is a pair $\mathcal{M} = (M, \mathcal{I})$ where M is a set of items and $\mathcal{I} \subseteq 2^M$ is a nonempty set of subsets of M (called *independent sets*) satisfying the following properties:
(i) *Downward-closed*: $S \subset T$ and $T \in \mathcal{I}$ implies $S \in \mathcal{I}$;
(ii) *Augmentation*: For every $S, T \in \mathcal{I}$, if $|S| < |T|$, then $S \cup \{g\} \in \mathcal{I}$ for some $g \in T$.
A *base* of $\mathcal{M}$ is a maximal independent set in $\mathcal{M}$.

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 $i, j \in N$.

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

Definition 2 (partition matroid)**.** A matroid $\mathcal{M} = (M, \mathcal{I})$ is a *partition matroid* if there exists a partition of M into *categories* $C = \{C^1, \dots, C^\ell\}$ for some $\ell \leq m$, and a corresponding vector of *capacities* $k^1, \dots, k^\ell$, such that the collection of independent sets is

$$\mathcal{I} = \{S \subseteq M : |S \cap C^h| \leq k^h \text{ for every } h \in [\ell]\}.$$

An instance with partition matroids is said to have *identical categories* if all the agents have the same partition into categories, i.e., $\ell_i = \ell_j = \ell$ and $C_i^h = C_j^h = C^h$ for all $i, j \in N$ and $h \in [\ell]$. The capacities, however, may be different.

Given an allocation $\mathbf{X}$, we denote by X_i^h the items from category C_i^h given to agent i in $\mathbf{X}$.

A special case of a partition matroid is a *uniform matroid*, which is a partition matroid with a single category.

Definition 3 (complete feasible allocation)**.** An allocation $\mathbf{X}$ is said to be :
(i) *feasible* — if $X_i \in \mathcal{I}_i$ for every agent i;
(ii) *complete* — if $\biguplus_i X_i = M$.

Let $\mathcal{F}$ denote the set of all complete feasible allocations. Throughout this paper we consider only instances that admit at least one complete feasible allocation:

Assumption 1. *All instances considered in this paper admit at least one complete feasible allocation; i.e., $\mathcal{F} \neq \emptyset$.*

With constraints based on partition matroids with identical categories, the assumption can be easily verified: it is satisfied if and only if, for every category C^h, the sum of agent capacities for this category is at least $|C^h|$. With constraints based on a common matroid, verifying the assumption is equivalent to the *matroid partition problem*, which can be solved in polynomial time (Edmonds, 1965). In fact, Assumption 1 can be verified in polynomial time even with different matroids: this follows from Theorem 9, which we prove in Subsection 7.1.

2.2 Valuations and Fairness Notions

Every agent i is associated with an *additive* valuation function $v_i : 2^M \to \mathbb{R}_+$, which assigns a positive real value to every set $S \subseteq M$.[2]

Additivity means that there exist m values $v_i(1), \dots, v_i(m)$ such that $v_i(S) = \sum_{g \in S} v_i(g)$. An additive valuation v_i is called *binary* if $v_i(g) \in \{0, 1\}$ for every $i \in N, g \in M$. An allocation $\mathbf{X}$ is *Social Welfare Maximizing* (SWM) if $\mathbf{X} \in \text{argmax}_{\mathbf{X}' \in \mathcal{F}} \sum_{i \in N} v_i(X'_i)$.

Definition 4 (envy and envy-freeness)**.** Given an allocation $\mathbf{X}$, agent i *envies* agent j if $v_i(X_i) < v_i(X_j)$. $\mathbf{X}$ is *envy-free (EF)* if no agent envies another agent.

Definition 5 (EF1)**.** (Budish, 2011) An allocation $\mathbf{X}$ is *envy-free up to one good (EF1)* if for every $i, j \in N$, there exists a subset $Z \subseteq X_j$ with $|Z| \leq 1$, such that $v_i(X_i) \geq v_i(X_j \setminus Z)$.

Definition 6. The set of *feasible subsets* of a set T for agent i, denoted $\text{FEAS}_i(T)$, is:

$$\text{FEAS}_i(T) := \{S \subseteq T : \ S \in \mathcal{I}_i\}$$

Definition 7 (feasible valuation)**.** The *feasible valuation* of agent i for a set T is denoted $\hat{v}_i(T)$ and defined as the maximum value of a subset of T that is feasible for i:

$$\hat{v}_i(T) := \max_{S \in \text{FEAS}_i(T)} v_i(S).$$

Definition 8. Given a feasible allocation $\mathbf{X}$:

- Agent i *F-envies* agent j iff $\hat{v}_i(X_i) < \hat{v}_i(X_j)$.
- $\mathbf{X}$ is *F-EF* (feasible-EF) if no agent F-envies another one.
- $\mathbf{X}$ is *F-EF1* if for every $i, j \in N$, there exists a subset $Z \subseteq X_j$ with $|Z| \leq 1$, such that $\hat{v}_i(X_i) \geq \hat{v}_i(X_j \setminus Z)$, that is, if Z is removed from j's bundle, then i does not F-envy j.

Note that, if all agents have the same constraints, then all bundles in a feasible allocation are feasible for all agents. Therefore, EF and F-EF are equivalent, and F-EF1 and EF1 are equivalent.

For further discussion of the F-EF1 criterion and some alternative (weaker) definitions, see Appendix A.

Another useful notation is *positive feasible-envy*, which is the amount by which an agent F-envies another agent:

Definition 9. The *positive feasible-envy* of agent i towards j in allocation $\mathbf{X}$ is:

$$\text{Envy}^+_{\mathbf{X}}(i, j) := \max(0, \hat{v}_i(X_j) - \hat{v}_i(X_i)).$$

Definition 10. For an allocation $\mathbf{X}$, the *envy-graph* $\mathcal{G}(\mathbf{X})$ is the 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 *feasible-envy-graph* is defined analogously based on the feasible-envy.

2. All our algorithms use a polynomial number of arithmetic operations. Thus, they run in polynomial time in any computational model in which arithmetic operations on the numbers representing agents' values can be done in constant time (e.g. the "unit cost" model for real numbers).

2.3 Common Tools and Techniques

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

Envy-cycle elimination. The first method for attaining an EF1 allocation (in unconstrained settings, even with arbitrary monotone valuations) is due to Lipton et al. (2004).

The *envy-cycle elimination* algorithm works as follows. Start with the empty allocation. Then, as long as there is an unallocated item: (i) choose an agent that is a source in the envy-graph (not envied by another agent) and give her an arbitrary unallocated item; (ii) reconstruct the envy-graph $\mathcal{G}$ corresponding to the new allocation; (iii) as long as $\mathcal{G}$ contains cycles, choose an arbitrary cycle, and shift the bundles along the cycle. This increases the total value, thus this process must end with a cycle-free graph.

Max Nash welfare. The *Nash Welfare* (NW) of an allocation $\mathbf{X}$ is the product of the agents' values: $NW = (\prod_{i \in N} v_i(X_i))$. An allocation is *max Nash welfare (MNW)* if it maximizes the NW among all feasible allocations. Caragiannis, Kurokawa, Moulin, Procaccia, Shah, and Wang (2019) showed that in unconstrained settings with additive valuations, every MNW allocation is EF1.

Round Robin (RR). RR works as follows. Given a fixed order σ over the agents, as long as there is an unallocated item, the next agent according to σ chooses an item she values most among the unallocated items (where the next of agent $\sigma[n]$ is agent $\sigma[1]$). Simple as it might be, this algorithm results in an EF1 allocation in unconstrained settings with additive valuations (Caragiannis et al., 2019).

Per category RR + envy-cycle elimination. This algorithm (Algorithm 1) was introduced by Biswas and Barman (2018) for finding an EF1 allocation when the constraints are based on identical partition matroids (where both the categories and the capacities are identical). It resolves the categories sequentially, resolving each one by RR followed by envy-cycle elimination, where the order over the agents is determined by a topological order in the obtained envy-graph.

ALGORITHM 1: Per-Category Round Robin (Biswas & Barman, 2018)

Initialize:
  $\sigma \leftarrow$ an arbitrary order over the agents.
  For all $i \in N$, set $X_i \leftarrow \emptyset$.
for every category h **do**
  Run Round Robin with C^h, σ.
  Let X_i^h be the resulting allocation for agent i.
  For all $i \in N$, let $X_i \leftarrow X_i \cup X_i^h$.
  Construct the envy-graph for the current allocation.
  Remove cycles from the graph, switching bundles along the cycles.
  Set σ to be a topological order of the graph.
end for
return the allocation $\mathbf{X}$.

2.4 Pareto-efficiency

Definition 11. A (feasible) allocation $\mathbf{X}$ is *Pareto-efficient* if there is no other feasible allocation $\mathbf{X}'$ such that all agents weakly prefer $\mathbf{X}'$ over $\mathbf{X}$ and at least one agent strictly prefers $\mathbf{X}'$.

We observe that, with additive identical valuations, every complete feasible allocation is Pareto-efficient.

Observation 1. *For any (possibly constrained) setting with identical additive valuations, every complete feasible allocation is Pareto-efficient.*

Proof. Let v denote the common valuation of all agents. Let $\mathbf{X}$ be a complete feasible allocation. Feasibility implies that for every agent $i \in N$: $\hat{v}_i(X_i) = v(X_i)$. Therefore, $\sum_{i\in N} \hat{v}_i(X_i) = \sum_{i\in N} v(X_i)$. By completeness and additivity, the latter sum equals $v(M)$, which is a constant independent of $\mathbf{X}$. So the sum of feasible values is the same in all complete feasible allocations. Therefore, if any other allocation gives a higher value to some agent, it must give a lower value to some other agent. $\square$

3. Partition Matroids with At Most Two Categories

In this section we initially assume that the agents' constraints are based on partition matroids with a single category (uniform matroids). This means that the constraint is only on the number of items given to each agent. We later extend the result to partition matroids with two categories.

3.1 Uniform Matroids

As a warm-up, we present a simple algorithm for a setting with a single category. We call it Capped Round Robin (CRR). CRR is a slight modification of Round Robin, where if an agent reached her capacity — she is skipped over (Algorithm 2).

ALGORITHM 2: Capped Round Robin

Input: Category C^h with capacities k_i^h for every $i \in N$, and an order σ over N.

Initialize: $L \leftarrow C^h$, $t \leftarrow 0$, $\forall i \in N : X_i^h \leftarrow \emptyset$.
while $L \neq \emptyset$ **do**
    $i \leftarrow \sigma[t]$.
    if $|X_i^h| < k_i^h$ **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
        $L \leftarrow L \setminus \{g\}$.
    end if
    $t \leftarrow t + 1 \mod n$.
end while
return X^h

CRR finds an F-EF1 allocation whenever the constraints of all agents are *uniform matroids*, i.e., partition matroids where all items belong to a single category (but agents may have heterogeneous capacities and heterogeneous valuations).

Theorem 2. *With uniform-matroid constraints, CRR finds an F-EF1 allocation using a polynomial number of arithmetic operations. Furthermore, if* $\mathbf{X}$ *is the outcome, for every* i, j *such that* i *precedes* j *in* σ, $v_i(X_i) = \hat{v}_i(X_i) \geq \hat{v}_i(X_j)$.

The proof is similar to that of standard Round Robin in unconstrained settings. We include it here for completeness.

Proof. Let $i, j \in N$ be two agents such that i precedes j in σ.

First, we prove that i does not envy j. Let Y_j be any feasible subset of X_j maximizing v_i, so that $\hat{v}_i(X_j) = v_i(Y_j)$.

Since i chooses first among i, j, $|X_i| \geq |Y_j|$. If we order X_i and Y_j according to the order in which the items were taken, every item in X_i was chosen before (and is therefore worth more to agent i than) the item in Y_j picked in the same round (if such one exist, as $|X_i| \geq |Y_j|$). That is because between the two of them, i chose first. So $v_i(X_i) \geq v_i(Y_j) = \hat{v}_i(X_j)$.

Now it remains to show that $\mathbf{X}$ satisfies the F-EF1 condition for j. Let g be the first item chosen by agent i. Notice that if we remove g from X_i, it is equivalent to j being the one choosing first among i, j (when the item set does not include g). Therefore, we can use the exact same argument to claim that $v_j(X_j) \geq \hat{v}_j(X_i \setminus \{g\})$.

CRR can be implemented by sorting the items in C^h, for each agent i, by descending order of v_i. This requires $O(n|C^h| \log(|C^h|))$ arithmetic operations. Then, the main loop requires $O(|C^h|)$ arithmetic operations. □

3.2 Two Categories

While CRR may not find an F-EF1 allocation for more than one category, we can extend it to two categories by running CRR with reverse order on the second category; see Algorithm 3.

Theorem 3. *When all agents have partition-matroid constraints with at most two categories, the same categories but possibly different capacities, an F-EF1 allocation always exists and can be found using a polynomial number of arithmetic operations.*

ALGORITHM 3: Back-and-Forth CRR

```
σ ← an arbitrary order over the agents.
Run Capped Round Robin with C^1, σ. Let X_i^1 be the outcome for each agent i ∈ N.
σ' ← reverse(σ).
Run Capped Round Robin with C^2, σ'. Let X_i^2 be the outcome for each agent i ∈ N.
return X_i^1 ∪ X_i^2 for all i ∈ N.
```

Proof. Algorithm *Back-and-forth CRR* (Algorithm 3) runs CRR in an arbitrary order for the first category, then uses the reverse order for CRR in the second category. After the first category, by Theorem 2, the allocation is F-EF1 and no agent envies another agent that appears in σ 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 σ' and thus will not gain any more envy in the second category. If i didn't F-envy j 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. We conclude that the resulting allocation is F-EF1.

Since CRR uses a polynomial number of arithmetic operations, the same is true for Algorithm 3. □

With three or more categories, when agents have additive heterogeneous valuations and heterogeneous constraints, it is open whether an EF1 allocation exist.

4. Partition Matroids with Heterogeneous Capacities, Identical Valuations

We now consider an arbitrary number of categories, allow agents to have heterogeneous capacities, but assume that all agents have the same valuations; this is, in a sense, the dual setting of Biswas and Barman (2018) who consider identical capacities and heterogeneous valuations. Using CRR as a subroutine, we show that a similar algorithm to the one used by Biswas and Barman (2018) finds an F-EF1 allocation in this setting; this follows from the fact that no cycles can be formed in the envy-graph. Using Algorithm 4 we prove:

Theorem 4. *For every instance with identical additive valuations and partition matroids with identical categories (but possibly different capacities), Algorithm 4 finds an F-EF1 allocation using a polynomial number of arithmetic operations.*

Similarly to Biswas and Barman (2018), our Algorithm 4 iterates over the categories running a sub-routine in each. While they run Round Robin, we run CRR. The order for the sub-routine is determined by a topological sort of the envy-graph. Biswas and Barman (2018) have an extra step of de-cycling the graph, which is not needed in our case due to the following lemma.

ALGORITHM 4: Per-Category Capped Round Robin

Input: M, C, k_i^h for every $i \in N, h \in [\ell]$
Output: an F-EF1 allocation $\mathbf{X}$.
Initialize: $\sigma \leftarrow$ an arbitrary order over the agents; $\forall i \in N\ X_i \leftarrow \emptyset$.
for all $C^h \in C$ **do**
    $\mathbf{X}^h \leftarrow$ Capped Round Robin on C^h with order σ.
    $\forall i \in N\ X_i \leftarrow X_i \cup X_i^h$.
    Set σ to be a topological order of the feasible-envy-graph (which is acyclic by Lemma 1).
end for
return the allocation $\mathbf{X}$.

Lemma 1. *Consider an instance with identical additive valuations and partition matroids with identical categories (but possibly different capacities). Let $\mathbf{X}$ be any feasible allocation. Then the feasible-envy-graph of $\mathbf{X}$ is acyclic.*

Proof. We first prove that the envy-graph of $\mathbf{X}$ is acyclic. Suppose by contradiction that there is an envy-cycle. W.l.o.g., suppose the cycle contains some $p \geq 2$ agents, with envy relations $1 \to 2 \to \cdots \to p \to 1$. Then, $v(X_1) < v(X_2) < \cdots v(X_p) < v(X_1)$, which is a contradiction.

Note that we cannot use the same argument directly for feasible-envy, since the feasible-valuation functions $\hat{v}_i$ may differ due to the different capacities. However, we can prove that the feasible-envy-graph is a subgraph of the envy-graph. Indeed, if some agent i F-envies some agent j, then $\hat{v}_i(X_i) < \hat{v}_i(X_j)$. But $\hat{v}_i(X_i) = v_i(X_i)$ since the allocation is feasible, and $\hat{v}_i(X_j) \leq v_i(X_j)$ by definition of feasible-valuation. Therefore, $v_i(X_i) < v_i(X_j)$, so i envies j.

Since the envy-graph has no cycles, every subgraph of it must have no cycles. □

Proof of Theorem 4. We show by induction that, after the assignment of all goods in each category, the allocation is F-EF1.

Base: After the first category, the allocation is F-EF1 by Theorem 2.

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 σ in Algorithm 2 (CRR). This is possible by Lemma 1, which shows that the feasible-envy-graph is acyclic. For every i, j such that i precedes j in σ, j does not F-envy i. By Theorem 2, during category $t+1$, j can become envious of i, but only up to one good, and i's envy cannot increase. This implies that if the allocation is F-EF1 in the end of category t, it remains F-EF1 after category $t+1$.

Since CRR uses a polynomial number of arithmetic operations, the same is true for Algorithm 4. □

The following theorem shows that, for identical valuations and possibly different capacities, the Maximum Nash Welfare allocation is F-EF1.

Theorem 5. *Suppose the constraints are based on partition matroids with identical categories but possibly different capacities. When agents have identical valuations, any feasible allocation that maximizes Nash Welfare is F-EF1.*

Proof. Let $\mathbf{X}$ be an allocation that maximizes Nash Welfare (MNW), and suppose there exist agents i, j such that i F-envies j. This means that, for at least one category h, $\hat{v}_i(X_i^h) < \hat{v}_i(X_j^h)$. Since the allocation is feasible, this implies $v(X_i^h) < \hat{v}_i(X_j^h)$. This implies $\hat{v}_i(X_j^h) > 0$, which implies $k_i^h > 0$ and $|X_j^h| > 0$. Without loss of generality, we can assume that $|X_i^h| = k_i^h$; otherwise, we can add to category h some $k_i^h - |X_i^h|$ dummy elements with value 0 to all agents and give them to agent i without affecting the valuations. Therefore, $|X_i^h| > 0$ too, so the following items exist:

$$b \in \operatorname*{argmin}_{t \in X_i^h} v(t); \qquad\qquad g \in \operatorname*{argmax}_{t \in X_j^h} v(t).$$

We have

$$\begin{aligned}
k_i^h \cdot v(g) &\geq \hat{v}_i(X_j^h) && \text{by definition of } g, \\
&> \hat{v}_i(X_i^h) && \text{since } i \text{ F-envies } j \text{ in category } C^h, \\
&= v(X_i^h) && \text{since } \mathbf{X} \text{ is feasible,} \\
&\geq k_i^h \cdot v(b) && \text{by definition of } b,
\end{aligned}$$

so $v(g) - v(b) > 0$.

Let $\mathbf{X}'$ be the allocation obtained from $\mathbf{X}$ by swapping goods b and g between i and j's allocations. $\mathbf{X}'$ is feasible since b and g are in the same category. Since $\mathbf{X}$ is MNW, it follows that

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

Let $z = v(g) - v(b) > 0$. 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

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

By the fact that $\hat{v}_i(S) \leq v(S)$ for every set S, we get

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

This implies that $\mathbf{X}$ is F-EF1, completing the proof. □

The fact that the constraints are based on partition matroids is used in the proof step regarding the exchange of items b and g. Possibly, the result can be extended to agents with different matroid constraints $(M, \mathcal{I}_i)$, as long as the different matroids satisfy some "pairwise basis exchange" property. So far, we could not formally define this property.

5. Partition Matroids, Binary Valuations

In this section we assume that all agents have *binary* additive valuations. For binary valuations $v_i(g) \in \{0, 1\}$ for all i, g. For every agent i we refer to the set of items $M_i = \{g \in M : v_i(g) = 1\}$ as agent i's *desired set*.

5.1 Algorithm for F-EF1 Allocation

We present an efficient algorithm that finds an F-EF1 allocation for n agents with heterogeneous binary valuations, and partition matroids with heterogeneous capacity constraints. Two key tools we use are the *agent-item graph* and the *priority matching*, defined next.

Definition 12 (Agent-item graph)**.** Given a category h and a partial allocation $\mathbf{X}$, the *agent-item graph* is a bipartite graph G^h, where one side consists of the agents with remaining capacity (i.e., agents such that $|X_i^h| < k_i^h$), and one side consists of the unallocated items of C^h. An edge (i, g) exists in G^h iff g is a desired item of i, that is, $v_i(g) = 1$.

Definition 13 (Priority matching)**.** Given a graph $G = (V, E)$, a matching in G is a subset of edges $\mu \subseteq E$ such that each vertex $u \in V$ is adjacent to at most one edge in μ. Given a linear order on the vertices, $\sigma[1], \ldots, \sigma[n]$, every matching is associated with a binary vector of size n, where element i equals 1 whenever vertex $\sigma[i]$ is matched. A *priority matching of* σ is a matching associated with the maximum such vector in the lexicographic order.

Note that every ordering σ over the vertices may yield a different priority-matching. Priority matchings were introduced by Roth, Sönmez, and Ünver (2005) in the context of kidney exchange, where they prove that every priority matching is also a *maximum-cardinality matching*; that is, it maximizes the total number of saturated vertices in V. Okumura (2014) extended this result to priority classes of arbitrary sizes, and showed a polynomial time algorithm for finding a priority matching. Simpler algorithms were presented by Turner (2015b, 2015a).

Theorem 6. *Suppose the constraints are based on partition matroids with identical categories but possibly different capacities. When agents have binary valuations, an F-EF1 allocation exists and can be computed in polynomial time by the* Iterated Priority Matching Algorithm *(Algorithm 5).*

ALGORITHM 5: Iterated Priority Matching

Initialize: $\forall i \in N : X_i \leftarrow \emptyset$.
for each category h **do**
  $\forall i \in N : X_i^h \leftarrow \emptyset$.

  `// Matching phase:`
  $T^h := \max_{i \in N} k_i^h$.
  for $t = 1, \ldots, T^h$ **do**
    Construct the agent-item graph G_t^h (see Definition 12).
    Construct the feasible-envy-graph corresponding to $\mathbf{X}$.
    $\sigma \leftarrow$ an ordering on the vertices of G_t^h where:
    (a) All agents are ordered before all items;
    (b) The ordering among the agents is a topological order on the feasible-envy-graph;
    (c) The ordering among the items is arbitrary.
    Find a priority matching in G_t^h according to σ (see Definition 13).
    For every agent i who is matched to an item g_i: $X_i^h \leftarrow X_i^h \cup \{g_i\}$.
  end for

  `// Leftover phase:`
  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 for
return the allocation $\mathbf{X}$.

The Iterated Priority Matching algorithm (Algorithm 5) works category-by-category. For each category h, the items of C^h are allocated in two phases, namely the *matching phase* and the *leftover phase*. The matching phase proceeds in several iterations, where in each iteration, every agent receives at most one item. The number of iterations is at most the maximum capacity of an agent in C^h, denoted by $T^h := \max_{i \in N} k_i^h$.

Given the current allocation, let σ be a topological order over the agents in the feasible-envy-graph (we shall soon show that the feasible-envy-graph is cycle free, so a topological order exists). In each iteration t of the matching phase, we construct the agent-item graph G_t^h. We extend σ to an ordering over all the vertices of G_t^h by ordering all the items, in an arbitrary order, below all the agents. We then compute a priority-matching in G_t^h with respect to σ, and add the obtained matches to the agent allocations. We then update the feasible-envy-graph and proceed to the next iteration, where the next set of items in C^h is allocated.

Finally, we allocate the leftover items arbitrarily among agents, respecting feasibility constraints. This is possible since a feasible allocation exists by assumption. We first prove that these leftover items are worth 0 for all agents with remaining capacity.

Lemma 2. *For every category h, all items allocated at the leftover phase are worth* 0 *for all agents with remaining capacity for C^h.*

Proof. For every category h and every agent i, in each iteration of the matching phase in which agent i has remaining capacity for C^h, at least one item valuable for i is allocated (either to i, or to some other agent, or both); otherwise, the matching could be augmented by assigning one such item to i, contradicting the maximality of the priority matching.

Therefore, after at most k_i^h iterations of the matching phase, there are only two options: either agent i has no remaining capacity (that is, i received exactly k_i^h items worth 1 for him), or agent i values all remaining items of C^h at 0.

Since $T^h := \max_{i \in N} k_i^h$, after at most T^h iterations of the matching phase, the same is true for all agents, that is, the remaining items are worthless for all agents with remaining capacity. □

To prove the correctness of the algorithm, it suffices to prove that every feasible-envy-graph constructed in the process is cycle-free, and that the feasible-envy between any two agents is at most 1. We prove both conditions simultaneously in the following lemma.

Lemma 3. *In every iteration of Algorithm 5:*

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

(b) For every $i, j \in N$, $\textsc{Envy}^+_{\mathbf{X}}(i,j) \leq 1$.

Proof. The proof is by induction on the categories and iterations. Both claims clearly hold from the outset (i.e., under the empty allocation). In the analysis below, we refer to states *before* (h,t) and *after* (h,t) to denote the states before and after iteration t of category h, respectively.

Proof of property (a). We assume that properties (a) and (b) both hold before $(h,1)$ (i.e., before starting to allocate items in category h). We prove that property (a) holds after (h,t) for every t. Suppose by contradiction that after (h,t) there is a cycle $i_1 \to \cdots \to i_p = i_1$ in the feasible-envy-graph. By the induction assumption (a), the cycle did not exist before category h, so at least one edge was created during the first t iterations in category h. Suppose w.l.o.g. that it is the edge $i_1 \to i_2$.

Let Q_1 be the set of items desired by i_1 that are allocated to i_1 up to (and including) iteration t of category h, and let $q = |Q_1|$. By Lemma 2, agent i_1 got these q items during the matching phase of h. Moreover, he must have gotten these q items in the first q iterations of the matching phase (otherwise, there exists an iteration $\leq q$ in which i_1 did not get an item, but a desired item remained unallocated, contradicting priority matching).

Let Q_2 be the set of items desired by i_1 that are allocated to i_2 up to iteration t of category h. The fact that i_1 started to F-envy i_2 during category h implies that $|Q_2| \geq q+1$

and $k^h_{i_1} \geq q+1$. By Lemma 2, agent i_2 got all these $q+1$ items during the matching phase of h. Moreover, he must have gotten these items in the first $q+1$ iterations of the matching phase (otherwise, one of these items could have been allocated to i_1 in iteration $q+1$, contradicting priority matching). Therefore, i_2 received at least $q+1$ items within the matching phase, implying that i_2's value increased by at least $q+1$ up to iteration t of category h.

Let Q_3 be the set of items desired by i_2 that are allocated to i_3 up to iteration t of category h. By assumption of the envy-cycle, i_2 F-envies i_3 after (h,t). By the induction assumption (b), $\textsc{Envy}^+_{\mathbf{X}}(i_2,i_3) \leq 1$ before $(h,1)$. Since i_2's value increased by at least $q+1$ up to iteration t of category h, it must hold that $|Q_3| \geq q+1$. We now claim that before $(h,q+1)$, at most one item of Q_3 was available, and i_3 got it in this iteration. Otherwise, one could allocate one of those items to i_2, and allocate the item that i_2 received in iteration $q+1$ (that is desired by i_1) to i_1, increasing the priority matching.

We conclude that i_3 got an item at each one of the first $q+1$ iterations of category h, as $|Q_3| \geq q+1$. Since all of these iterations are within the matching phase, all of these items are desired by i_3. Therefore, i_3's value increases by at least $q+1$. Repeating this argument, we conclude that every agent along the cycle received at least $q+1$ desired items during the first t steps of h, including agent $i_p = i_1$; but this is in contradiction to the fact that i_1 received only $q = |Q_1|$ items.

Proof of property (b). We assume that property (b) holds for every iteration before (h,t) and prove that it holds after (h,t).

Let i_1, i_2 be arbitrary agents. By the induction assumption, before $(h,1),\ldots,(h,t)$ we had $\textsc{Envy}^+_{\mathbf{X}}(i_1,i_2) \leq 1$. We consider several cases.

Case 1: Before (h,t), we had $\textsc{Envy}^+_{\mathbf{X}}(i_1,i_2) = 0$. Since at most one item is allocated to i_2 at iteration t, we must have $\textsc{Envy}^+_{\mathbf{X}}(i_1,i_2) \leq 1$ after (h,t).

Case 2: Before (h,t), the capacity of agent i_1 was exhausted, so we have $v_{i_1}(X^h_{i_1}) = k^h_{i_1} \geq \hat{v}_{i_1}(X^h_{i_2})$. But before $(h,1)$ we had $\textsc{Envy}^+_{\mathbf{X}}(i_1,i_2) \leq 1$, and the envy cannot increase after adding $k^h_{i_1}$ to $v_{i_1}(X_{i_1})$ and at most $k^h_{i_1}$ to $\hat{v}_{i_1}(X_{i_2})$.

Case 3: Agent i_1 does not desire any item remaining before (h,t). Then clearly the envy of i_1 cannot change during (h,t).

The remaining case is that, before (h,t) we had $\textsc{Envy}^+_{\mathbf{X}}(i_1,i_2) = 1$, the capacity of i_1 was not exhausted, and i_1 desires at least one item. Then, i_1 precedes i_2 in the topological order σ in iteration t, so the priority-matching on G^h_t prefers to match i_1, than to match i_2 and leave i_1 unallocated. Therefore, the envy of i_1 at i_2 does not increase during (h,t).

Finally, we claim that properties (a) and (b) are not affected by the leftover phase. By Lemma 2, for every agent i and category h, there are two options: either agent i has received exactly k^h_i desirable items of category h, or agent i values all leftover items at 0. In the former case, the F-envy of i at any other agent cannot increase during category h at all. In the latter case, the F-envy of i cannot increase during the leftover phase of category h. In any case, the leftover phase cannot add new envy edges or increase the envy levels. Therefore, properties (a) and (b) still hold after the leftover phase. □

5.2 Pareto-efficiency

We show that if capacities are binary (that is, $k_i^h \in \{0,1\}$ for all i, h), then Algorithm 5 returns a Pareto-efficient allocation, but this is not the case under arbitrary (non-binary) capacities.

Observation 7. *In settings with partition constraints with heterogeneous* binary *capacities and heterogeneous binary valuations, Algorithm 5 returns a Pareto-efficient allocation.*

Proof. Under binary capacities, the algorithm runs a single priority matching in each category. As this matching is of maximum cardinality, it maximizes the social welfare within this category. From additivity, the allocation that maximizes social welfare within every category maximizes social welfare over all categories. Any allocation maximizing the social welfare is Pareto-efficient. □

The following example shows that when capacities may be larger than 1, even when there is a single category and two agents with the same capacity, the allocation returned by Algorithm 5 may not be Pareto-efficient.

Example 1. Consider the setting depicted in the following table. The agents are Alice and Bob, with valuations v_A and v_B respectively. The allocation is represented by asterisks.

Item	x_1	y_1	y_2	z_1
v_A	1 *	1 *	1	0
v_B	1	0	0 *	1 *

There are two agents sharing an identical uniform matroid with capacity 2, and four items. We claim that the allocation depicted in the table can be the outcome of Algorithm 5, and is not Pareto-efficient. Indeed, in the first iteration the priority matching may assign x_1 to Alice and z_1 to Bob. Then, Bob does not want any remaining item, so in the second iteration Alice gets y_1. Finally, in the leftover phase Bob gets y_2. In the obtained allocation, Alice has value 2 and Bob has value 1. This allocation is Pareto-dominated by the allocation giving y_1, y_2 to Alice and x_1, z_1 to Bob: Alice is indifferent and Bob is strictly better off. □

It remains open whether the setting with binary valuations and general heterogeneous capacities always admits an allocation that is both F-EF1 and Pareto-efficient.

6. Partition Matroids, Two Agents

In this section we present an algorithm for two agents with heterogeneous capacities. To present the algorithm we introduce some notation.

- Given an allocation $\mathbf{X}$, the *surplus* of agent i in category h is

  $$\text{SURP}_i^h(\mathbf{X}) := \hat{v}_i(X_i^h) - \hat{v}_i(X_j^h).$$

  That is, the difference between i's value for her own bundle and her value for j's bundle.

- Given agents $1, 2$, valuation functions v, v', an integer $a \in \{1, 2\}$ and a category h, $\mathcal{R}(v, v'; a)^h$ denotes the allocation obtained by Capped Round Robin (Algorithm 2 in Section 3) for category h, under valuations $v_1 = v, v_2 = v'$, and where agent a plays first. When the category is clear from the context, we omit the superscript h.

Theorem 8. *In every setting with two agents and partition matroid constraints with identical categories (but possibly different capacities), an F-EF1 allocation exists and can be computed by Algorithm "Round Robin Squared" (RR^2, Algorithm 6) using a polynomial number of arithmetic operations.*

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 by descending order of the surplus $\textsc{surp}_i^h(\mathcal{R}(v_1, v_2; i))$, that is, the surplus that agent i can gain over the other agent by playing first in category h. Denote this order π_i.

In the first iteration, agent 1 chooses the first category in π_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 π_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 surplus-order that has not been chosen yet, and within that category, agents are allocated according to CRR with that agent choosing first. This proceeds until all categories are allocated.

ALGORITHM 6: RR-Squared (RR^2)

Input: A set of items M, categories $C^1, \ldots, C^\ell$, capacities k_i^h for every $i \in \{1, 2\}, h \in [\ell]$; $a \in \{1, 2\}$ the first agent to choose.

Initialize for all $i \in \{1, 2\}$: $X_i \leftarrow \emptyset$; $\pi_i \leftarrow$ Categories listed by descending order of $\textsc{surp}_i^h(\mathcal{R}(v_1, v_2; i))$.
while there are unallocated categories **do**
    $h \leftarrow$ the first category in π_a not yet allocated.
    Run CRR on category h. Let $X^h \leftarrow \mathcal{R}(v_1, v_2; a)^h$.
    For all $i \in \{1, 2\}$, let $X_i \leftarrow X_i \cup X_i^h$.
    Switch a to be the other agent.
end while
return the allocation $\mathbf{X}$.

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. Formally:

Lemma 4. *For every category h:*

$$(a) \quad \textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)^h) \geq -\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 2)^h);$$
$$(b) \quad \textsc{surp}_2^h(\mathcal{R}(v_1, v_2; 2)^h) \geq -\textsc{surp}_2^h(\mathcal{R}(v_1, v_2; 1)^h).$$

The proof is based on Lemmas 5 through 7, which consider a setting with two agents playing CRR (Algorithm 2) on a single category.

We introduce the following notation for the proofs. Given a valuation function v and an item j, we denote:

- $\#[v > j] := |\{j' \in M : v(j') > v(j)\}|$ = the number of items more valuable than j;
- $\#[v \geq j] := |\{j' \in M : v(j') \geq v(j)\}|$ = the number of items at least as valuable as j (including j itself).

Observation 1. *If $\#[v \geq j] > \#[v > k]$ then $v(j) \leq v(k)$.*

Proof. By contradiction: if $v(j) > v(k)$ then the set of items more valuable than k includes j and all the items that are at least as valuable as j, and therefore $\#[v > k] \geq \#[v \geq j]$. $\square$

Lemma 5. *The worst case for each agent i with true valuation v_i is that the other agent plays according to the same valuation v_i. Formally, for every $a \in \{1, 2\}$ and valuation v':*

$$\begin{aligned}
&(a) \quad \hat{v}_2(\mathcal{R}(v_2, v_2; a)_2) \leq \hat{v}_2(\mathcal{R}(v', v_2; a)_2) \\
&(b) \quad \hat{v}_1(\mathcal{R}(v_1, v_1; a)_1) \leq \hat{v}_1(\mathcal{R}(v_1, v'; a)_1)
\end{aligned}$$

Proof. The two statements are obviously analogous; below we prove claim (a). Denote:

- $\gamma := \mathcal{R}(v_2, v_2; a)_2$ ordered by descending v_2, such that $v_2(\gamma_1) \geq v_2(\gamma_2) \geq \cdots$;
- $\gamma' := \mathcal{R}(v', v_2; a)_2$ ordered by descending v_2, such that $v_2(\gamma'_1) \geq v_2(\gamma'_2) \geq \cdots$.

Ties between items with same v_2 are broken arbitrarily but consistently (e.g. by item name).

Note that $|\gamma| = |\gamma'|$, as agent 2 gets the same number of items in both plays (the number of items given to agent 2 depends only on the agents' capacities and not on their valuations). We have to prove $\hat{v}_2(\gamma) \leq \hat{v}_2(\gamma')$. We prove a stronger claim: $v_2(\gamma_t) \leq v_2(\gamma'_t)$ for all $t \leq |\gamma|$.

For every index $t \leq |\gamma|$, denote by z_t the number of items held by agent 1 in round t of agent 2, that is:

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

When both agents play according to v_2, after agent 2 picks γ_t, agents 1 and 2 together hold the $z_t + t$ most-valuable (by v_2) items; hence, $\#[v_2 \geq \gamma_t] \geq z_t + t$.

When agent 1 plays according to v', agent 2 still plays by v_2 and thus takes γ'_t at round t (assuming consistent tie-breaking), while holding $t - 1$ items that are more valuable (or as valuable as) γ'_t. At that point, agent 1 holds at most z_t items more valuable than γ'_t. All in all, $\#[v_2 > \gamma'_t] \leq z_t + t - 1$.

Combining the two inequalities gives $\#[v_2 \geq \gamma_t] > \#[v_2 > \gamma'_t]$, which by Observation 1 gives $v_2(\gamma'_t) \geq v_2(\gamma_t)$ as claimed. $\square$

The next lemma considers the value each agent attributes to the other agent's bundle:

Lemma 6. *Every agent i with true valuation v_i weakly prefers the bundle the other agent gets when playing according to v_i, over the bundle the other agent gets when playing according to any other valuation. Formally, for every $a \in \{1, 2\}$ and every valuation v':*

$$\begin{aligned}
&(a) \quad \hat{v}_1(\mathcal{R}(v_1, v_1; a)_2) \geq \hat{v}_1(\mathcal{R}(v_1, v'; a)_2) \\
&(b) \quad \hat{v}_2(\mathcal{R}(v_2, v_2; a)_1) \geq \hat{v}_2(\mathcal{R}(v', v_2; a)_1)
\end{aligned}$$

Proof. The two statements are analogous; we prove claim (b). As in the proof of Lemma 5, we denote $\gamma := \mathcal{R}(v_2, v_2; a)_1$ ordered by descending v_2 and $\gamma' := \mathcal{R}(v', v_2; a)_1$ ordered by descending v_2, and prove $v_2(\gamma_t) \geq v_2(\gamma'_t)$ for all $t \leq |\gamma|$. Now we denote by z_t the number of items held by agent 2 in round t of agent 1:

$$z_t := \begin{cases} \min(k_2^h, t-1) & \text{if agent 1 plays first } (a=1) \\ \min(k_2^h, t) & \text{if agent 2 plays first } (a=2) \end{cases}$$

When both agents play according to v_2, before agent 1 picks γ_t, agents 1 and 2 together hold the $z_t + t - 1$ most-valuable (by v_2) items; hence, $\#[v_2 > \gamma_t] \leq z_t + t - 1$.

When agent 1 plays according to v', agent 2 still plays by v_2 and thus takes at least z_t of the $z_t + t - 1$ most-valuable items. By definition of the order γ', agent 1 holds at least t items that are at least as valuable as γ'_t. All in all, $\#[v_2 \geq \gamma'_t] \geq z_t + t$.

Combining the two inequalities gives $\#[v_2 \geq \gamma'_t] > \#[v_2 > \gamma_t]$, which by Observation 1 gives $v_2(\gamma_t) \geq v_2(\gamma'_t)$ as claimed. □

The next lemma considers a situation when both agents play by the same valuation:

Lemma 7. *Suppose both agents play according to the same valuation.*

(a) The value of each agent for her own bundle when she plays first is at least her value for the other agent's bundle when the other agent plays first:

$$\hat{v}_1(\mathcal{R}(v_1, v_1; 1)_1) \geq \hat{v}_1(\mathcal{R}(v_1, v_1; 2)_2)$$
$$\hat{v}_2(\mathcal{R}(v_2, v_2; 2)_2) \geq \hat{v}_2(\mathcal{R}(v_2, v_2; 1)_1)$$

(b) Similarly, the value of each agent for her own bundle when the other agent plays first is at least her value for the other agent's bundle when she plays first:

$$\hat{v}_1(\mathcal{R}(v_1, v_1; 2)_1) \geq \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_2)$$
$$\hat{v}_2(\mathcal{R}(v_2, v_2; 1)_2) \geq \hat{v}_2(\mathcal{R}(v_2, v_2; 2)_1)$$

Proof. We prove both claims for agent 1 only; the proof for agent 2 is analogous.

(a) When both agents play using the same valuation, the only thing that differentiates agent 1's bundle when 1 chooses first from agent 2's bundle when 2 chooses first is their capacities.

If $k_1^h \leq k_2^h$, then $\mathcal{R}(v_1, v_1; 1)_1 \subseteq \mathcal{R}(v_1, v_1; 2)_2$, and $\mathcal{R}(v_1, v_1; 1)_1$ contains the k_1^h highest-valued items in $\mathcal{R}(v_1, v_1; 2)_2$. Hence, $\mathcal{R}(v_1, v_1; 1)_1$ maximizes v_1 within $\text{FEAS}_1(\mathcal{R}(v_1, v_1; 2)_2)$, so $\hat{v}_1(\mathcal{R}(v_1, v_1; 1)_1) = \hat{v}_1(\mathcal{R}(v_1, v_1; 2)_2)$. Therefore, (a) holds with equality.

If $k_2^h \leq k_1^h$, then $\mathcal{R}(v_1, v_1; 2)_2 \subseteq \mathcal{R}(v_1, v_1; 1)_1$, and both these bundles are feasible for agent 1. Therefore, $\hat{v}_1(\mathcal{R}(v_1, v_1; 1)_1) \geq \hat{v}_1(\mathcal{R}(v_1, v_1; 2)_2)$, establishing (a).

(b) is proved using similar arguments.

If $k_1^h \leq k_2^h$, then $\mathcal{R}(v_1, v_1; 2)_1 \subseteq \mathcal{R}(v_1, v_1; 1)_2$, and $\mathcal{R}(v_1, v_1; 2)_1$ contains the k_1^h highest-valued items in $\mathcal{R}(v_1, v_1; 1)_2$. Hence, $\mathcal{R}(v_1, v_1; 2)_1$ maximizes v_1 within $\text{FEAS}_1(\mathcal{R}(v_1, v_1; 1)_2)$, so $\hat{v}_1(\mathcal{R}(v_1, v_1; 2)_1) = \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_2)$. Therefore, (b) holds with equality.

If $k_2^h \leq k_1^h$, then $\mathcal{R}(v_1, v_1; 1)_2 \subseteq \mathcal{R}(v_1, v_1; 2)_1$, and both bundles are feasible for agent 1. Therefore, $\hat{v}_1(\mathcal{R}(v_1, v_1; 2)_1) \geq \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_2)$, establishing (b). □

With these lemmas in hand, we are ready to prove Lemma 4.

Proof of Lemma 4. We provide the proof for the first statement:

$$\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)^h) \geq -\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 2)^h).$$

The proof of the second statement is analogous.

$$\begin{aligned}
&\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)) \\
&= \hat{v}_1(\mathcal{R}(v_1, v_2; 1)_1) - \hat{v}_1(\mathcal{R}(v_1, v_2; 1)_2) \\
&\geq \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_1) - \hat{v}_1(\mathcal{R}(v_1, v_2; 1)_2) && \text{Lemma 5(b) with } v' = v_2 \text{ and } a = 1. \\
&\geq \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_1) - \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_2) && \text{Lemma 6(a) with } v' = v_2 \text{ and } a = 1. \\
&\geq \hat{v}_1(\mathcal{R}(v_1, v_1; 2)_2) - \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_2) && \text{Lemma 7(a) for agent 1.} \\
&\geq \hat{v}_1(\mathcal{R}(v_1, v_2; 2)_2) - \hat{v}_1(\mathcal{R}(v_1, v_1; 1)_2) && \text{Lemma 6(a) with } v' = v_2 \text{ and } a = 2. \\
&\geq \hat{v}_1(\mathcal{R}(v_1, v_2; 2)_2) - \hat{v}_1(\mathcal{R}(v_1, v_1; 2)_1) && \text{Lemma 7(b) for agent 1.} \\
&\geq \hat{v}_1(\mathcal{R}(v_1, v_2; 2)_2) - \hat{v}_1(\mathcal{R}(v_1, v_2; 2)_1) && \text{Lemma 5(b) with } v' = v_2 \text{ and } a = 2. \\
&= -\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 2)).
\end{aligned}$$

□

Finally, we use Lemma 4 to prove Theorem 8.

Proof of Theorem 8. Without loss of generality, suppose agent 1 is the first to choose a category. Let $C^1, \ldots, C^\ell$ be the categories in the order they are chosen. If ℓ is odd, we add a dummy empty category to make it even. We show first that agent 1 does not F-envy agent 2. We have

$$\begin{aligned}
v_1(X_1) - \hat{v}_1(X_2) &= \sum_{h \in \{1,\ldots,\ell\}} v_1(X_1^h) - \sum_{h \in \{1,\ldots,\ell\}} \hat{v}_1(X_2^h) && \text{(by additivity)} \\
&= \sum_{h \in \{1,\ldots,\ell\}} (v_1(X_1^h) - \hat{v}_1(X_2^h)) \\
&= \sum_{h \in \{1,\ldots,\ell\}} \textsc{surp}_1^h(X) && \text{(by definition of surplus)} \\
&= \sum_{h \text{ is odd}} \textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)) + \sum_{h \text{ is even}} \textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 2)) && \text{(by the Round Robin order)} \\
&\geq \sum_{h \text{ is odd}} \textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)) + \sum_{h \text{ is even}} -\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)) && (1) \\
&= \sum_{t \in \{1,\ldots,\frac{\ell}{2}\}} (\textsc{surp}_1^{2t-1}(\mathcal{R}(v_1, v_2; 1)) - \textsc{surp}_1^{2t}(\mathcal{R}(v_1, v_2; 1))). && (2)
\end{aligned}$$

Equation (1) is attained by applying Lemma 4 to the rightmost summand: $\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)^h) \geq -\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 2)^h)$ is equivalent to $\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 2)^h) \geq -\textsc{surp}_1^h(\mathcal{R}(v_1, v_2; 1)^h)$. Since agent 1 chooses the odd categories, and does so based on highest surplus, it implies that for every t, $\textsc{surp}_1^{2t-1}(\mathcal{R}(v_1, v_2; 1)) \geq \textsc{surp}_1^{2t}(\mathcal{R}(v_1, v_2; 1))$, as category $2t$ was available when agent 1 chose category $2t-1$. Therefore, every summand in the sum of (2) is non-negative. Thus, the whole sum is non-negative, implying that $v_1(X_1) \geq \hat{v}_1(X_2)$, as claimed.

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 one 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 feasible-envy altogether. We conclude that the obtained allocation is F-EF1.

Algorithm 6 executes CRR a polynomial number of times, and each execution uses a polynomial number of arithmetic operations by Theorem 2. Therefore, Algorithm 6 too requires a polynomial number of arithmetic operations. □

The existence of EF1 allocations for three or more agents with heterogeneous capacities remains open. For two agents, the notion of surplus allows us to treat each category as a single item, whose value for agent i is $\text{SURP}_i^h(\mathcal{R}(v_1, v_2; i)^h)$. The challenge in extending our approach to $n \geq 3$ agents is that, for each category, there are $n!$ possible Round Robin orders, so it is not clear how to extend the notion of "surplus".

7. Base-Orderable Matroids, At Most Three Agents

In this section we consider constraints that are represented by a wide class of matroids, termed *base-orderable* (BO) matroids.[3] Recall that the *bases* of a matroid are its inclusion-maximal independent sets. In the definitions below, we use the shorthands $S+x := S \cup \{x\}$ and $S-x := S \setminus \{x\}$, for any set S and item x.

Definition 14. Given a matroid $(M, \mathcal{I})$ and independent sets $I, J \in \mathcal{I}$, a pair of items $x \in I$ and $y \in J$ is called a *feasible swap* if both $I - x + y$ and $J - y + x$ are in $\mathcal{I}$.

Definition 15. (Brualdi & Scrimger, 1968). A matroid $\mathcal{M} = (M, \mathcal{I})$ is *base-orderable (BO)* if for every two bases $I, J \in \mathcal{I}$, there exists a *feasible exchange bijection*, defined as a bijection $\mu : I \to J$ such that the pair $(x, \mu(x))$ is a feasible swap for all $x \in I$.

This class contains many interesting matroids, including partition matroids, laminar matroids (a natural generalization of partition matroids where the categories may be partitioned into sub-categories),[4] transversal matroids,[5] and more.

Algorithms on general matroids usually get access to an input matroid $\mathcal{M}$ through an *oracle*. A common type of oracle is an *independence oracle* — a Boolean function that

3. The class of base-orderable matroids was first introduced by Brualdi and Scrimger (1968), Brualdi (1969), but the term base-orderable appeared only later (Brualdi, 1971).
4. Formally, a *laminar matroid* is defined using ℓ possibly-overlapping sets $C_1, C_2, \ldots, C_\ell \subseteq M$ such that $\bigcup_{h=1}^{\ell} C_h = M$. Additionally, for every pair $h, h' \in \{1, \ldots, \ell\}$, only one of the following holds: (i) $C_h \subset C_{h'}$ or, (ii) $C_{h'} \subset C_h$ or, (iii) $C_h \cap C_{h'} = \emptyset$. Each C_h has a capacity k_h. A set $I \subseteq M$ is independent if and only if $|I \cap C_h| \leq k_h$ for each $h \in \{1, \ldots, \ell\}$.
5. Given a bipartite graph $G = (L+R, E)$, the *transversal matroid* is a matroid on L. Its independent sets are all subsets of L which are sets of endpoints of matchings in G. Formally, $\mathcal{I} = \{A \subseteq L :$ there exists an injective function $f : A \to R$ such that, for all $a \in A$, it holds that $(a, f(a)) \in E\}$.

decides whether a given set of elements is an independent set of $\mathcal{M}$ (Rado, 1942). Using this oracle, it is possible to construct a *circuit-finding oracle*, that returns, for every set that is not independent in $\mathcal{M}$, a *circuit* — a minimal dependent subset (Robinson & Welsh, 1980).

Lemma 8. *Let $\mathcal{M}$ be a base-orderable matroid. For every two bases I, J of $\mathcal{M}$, it is possible to find a feasible-exchange bijection using a polynomial number of arithmetic operations and calls to an independence oracle of $\mathcal{M}$.*

Proof. Construct a bipartite graph G with the elements of I on one side and the elements of J on the other side. For an element in both I and J, there are two different vertices — one on each side. Add an edge between elements $x \in I$ and $y \in J$ whenever $I - x + y$ is a basis. This can be checked using the independence oracle, and therefore G can be constructed in polynomial time.

Find a perfect matching in G; such a matching exists since $\mathcal{M}$ is base-orderable. Return a feasible-exchange bijection μ that maps each element in I to the element matched to it in J.[6] □

When different agents have different matroids, even when these are all partition matroids, an F-EF1 allocation may not exist (see Example 4 in Section 8.2 below). Therefore, we restrict attention to settings with a common matroid $\mathcal{M}$. With identical constraints, F-EF1 and EF1 are equivalent, so in this section we only mention EF1. Before presenting our algorithm, we present two tools that are useful for any matroid — BO or not: finding an allocation maximizing the social welfare with matroid constraints (Section 7.1) and extending a matroid such that every feasible partition is a partition into bases (Section 7.2).

7.1 Finding an Allocation Maximizing the Social Welfare

We initialize our algorithm with an allocation that maximizes the sum of agents' utilities.

Such an allocation can be found in polynomial time even for heterogeneous matroid constraints.

Theorem 9. *Suppose there are any n agents with additive valuations, where the bundle of each agent i must be an independent set of a matroid $\mathcal{M}_i$. It is possible to find, using a polynomial number of arithmetic operations, an allocation that maximizes the sum of utilities among all complete feasible allocations, or decide that no complete feasible allocation exists.*

Proof. The problem can be reduced to the *maximum-weight matroid intersection* problem:[7] given two matroids over the same base-set, $(Z, \mathcal{I}_a)$ and $(Z, \mathcal{I}_b)$, where each element of Z has a weight, find an element of $\mathcal{I}_a \cap \mathcal{I}_b$ with a largest total weight (Edmonds (1970), statement (46)). There are many polynomial-time algorithms for weighted matroid intersection. For example, an algorithm by Frank (1981) solves the problem using $O(T|Z|^3)$ arithmetic operations, where T is the number of operations required by circuit-finding oracles for the input matroids.

6. We are grateful to Joshua Grochow for the proof idea.
7. We are grateful to Jan Vondrak and Chandra Chekuri for the proof idea. A similar idea was used by Lehmann, Lehmann, and Nisan (2001), Vondrák (2008) and Calinescu, Chekuri, Pál, and Vondrák (2011)[section 4.1] for approximate welfare maximization with submodular utilities.

We construct the base set $Z := N \times M$, where each pair $(i, g) \in Z$ corresponds to allocating item $g \in M$ to agent $i \in N$. We construct two matroids over Z, namely $\mathcal{M}_a = (Z, \mathcal{I}_a)$ and $\mathcal{M}_b = (Z, \mathcal{I}_b)$. We first describe the independent sets in both matroids, then specify the weight function.

The first matroid, $\mathcal{M}_a$, is a partition matroid with m categories, where category g corresponds to the set $\{(1, g), \dots, (n, g)\}$, and every category has capacity 1. Membership in $\mathcal{M}_a$ ensures that every item is given to at most one agent.

The second matroid, $\mathcal{M}_b$, represents the matroid constraints of the n agents in the fair allocation problem. For every subset $S \subseteq Z$, denote $A_i(S) := \{g : (i, g) \in S\}$, which represents the set of items allocated to i by S. Then S is in $\mathcal{I}_b$ iff for every agent i, the set $A_i(S)$ is an independent set in $\mathcal{M}_i$. We show that $\mathcal{M}_b$ is a matroid:

(i) Downward-closed: suppose $S \subset T$ and $T \in \mathcal{I}_b$. For all $i \in N$, $S \subset T$ implies $A_i(S) \subseteq A_i(T)$, and $T \in \mathcal{I}_b$ implies $A_i(T) \in \mathcal{I}_i$. Since $\mathcal{M}_i$ is downward-closed, $A_i(S) \in \mathcal{I}_i$. By the definition of $\mathcal{I}_b$ we conclude that $S \in \mathcal{I}_b$.

(ii) Augmentation property: Let $S, T \in \mathcal{I}_b$ such that $|S| < |T|$. Since $|S| = \sum_{i \in N} |A_i(S)|$ and $|T| = \sum_{i \in N} |A_i(T)|$, there must be some $i \in N$ for which $|A_i(S)| < |A_i(T)|$. Since $A_i(S), A_i(T) \in \mathcal{I}_i$, the augmentation property of $\mathcal{M}_i$ implies that there exists an item $g \in A_i(T) \setminus A_i(S)$ such that $A_i(S) \cup \{g\} \in \mathcal{I}_i$. Then $S \cup \{(i, g)\} \in \mathcal{I}_b$ and the augmentation property holds.

Given independence oracles for the matroids $\mathcal{M}_i$, for all $i \in N$, one can construct an independence oracle for $\mathcal{M}_b$ as follows: run the independence oracle of every $\mathcal{M}_i$ on the set $A_i(S)$, and return "true" if and only if all n independence oracles returned "true".

One can easily verify that every subset of Z that is an independent set in both $\mathcal{M}_a$ and $\mathcal{M}_b$ represents a feasible allocation.

Next, define the weight function $w : Z \to \mathbb{R}_+$. For every $(i, g) \in Z$, let $w((i, g)) = V + v_i(g)$, where $V := 2m \cdot \max_i \max_g v_i(g)$. Find a maximum-weight subset $S^* \in \mathcal{I}_a \cap \mathcal{I}_b$. The construction of w guarantees that S^* maximizes the number of allocated items. Subject to this, S^* maximizes the total value.

If the number of allocated items is smaller than the total number of items, then we can conclude that no complete feasible allocation exists. Otherwise, a complete feasible allocation is found. Within complete allocations, maximizing the total value ensures that the returned allocation maximizes social welfare. This concludes the proof. □

7.2 Ensuring a Partition into Bases

To simplify the algorithms, we pre-process the instance to ensure that, in all feasible allocations, every agent receives a *basis* of $\mathcal{M}$. Recall that the *rank* of a matroid $\mathcal{M}$ is the cardinality of a basis of $\mathcal{M}$ (all bases have the same cardinality). We denote $r := \text{rank}(\mathcal{M})$. In the pre-processing step, we add to M dummy items, valued at 0 by all agents, so that after the addition, $|M| = n \cdot r$. This guarantees that, in every feasible allocation, every bundle contains exactly r items, so it is a basis. To ensure that the dummy items do not affect the set of feasible allocations, we use the *free extension*,[8] defined below.

8. We are grateful to Kevin Long for the proof idea at https://math.stackexchange.com/q/4300433.

Definition 16. Let $\mathcal{M} = (M, \mathcal{I})$ be a matroid with rank r. The *free extension* of $\mathcal{M}$ is a matroid $\mathcal{M}' = (M', \mathcal{I}')$ defined as follows (where x^{new} is a new item):

$$\begin{aligned} M' &:= M + x^{\text{new}}; \\ \mathcal{I}' &:= \mathcal{I} \quad \cup \quad \{I + x^{\text{new}} : \ I \in \mathcal{I}, |I| \leq r - 1\}. \end{aligned}$$

That is: all bundles that were previously feasible remain feasible; in addition, all non-maximal feasible bundles remain feasible when the new item is added to them.

The properties of the free extension are summarized below.

Observation 2. *If $\mathcal{M}'$ is the free extension of $\mathcal{M}$ with new item x^{new}, then:*

- *All bases of $\mathcal{M}$ are bases of $\mathcal{M}'$.*
- $\text{rank}(\mathcal{M}) = \text{rank}(\mathcal{M}') = r$.
- *Given a feasible partition of M (a partition into independent sets), where some sets in the partition are not maximal (contain less than r items), one can construct a feasible partition of M' by adding x^{new} into some non-maximal set.*
- *Given a feasible partition of M', one can construct a feasible partition of M by removing x^{new} from the set containing it.*
- *$\mathcal{M}'$ is base-orderable if and only if $\mathcal{M}$ is base-orderable.*

The first four observations are trivial. We prove the fifth one in Appendix B.

By Assumption 1, our instance admits a feasible allocation. Since any feasible bundle is contained in a basis, the cardinality of every allocated bundle is at most r, so $|M| \leq n \cdot r$. We construct a new instance by applying the free extension $n \cdot r - |M|$ times, getting a matroid $\mathcal{M}' = (M', \mathcal{I}')$ with $|M'| = n \cdot r$. We call the $n \cdot r - |M|$ new items *dummy items*, and let all agents value them at 0.

Observation 3. *The new instance satisfies the following properties.*

- *All bases of $\mathcal{M}$ are bases of $\mathcal{M}'$.*
- *In every feasible allocation $(Y_1, \ldots, Y_n)$ in the new instance, $|Y_i| = r$ for all $i \in N$, so every Y_i is a basis of $\mathcal{M}'$.*
- *For every feasible allocation $(X_1, \ldots, X_n)$ in the original instance, there is a feasible allocation $(Y_1, \ldots, Y_n)$ in the new instance, where for all $i \in N$, Y_i contains X_i plus zero or more dummy items, so $v_i(X_j) = v_i(Y_j)$ for all $i, j \in N$.*
- *For every feasible allocation $(Y_1, \ldots, Y_n)$ in the new instance, there is a feasible allocation $(X_1, \ldots, X_n)$ in the original instance, where for all $i \in N$, Y_i contains X_i plus zero or more dummy items, so all agents' valuations to all bundles are identical.*
- *$\mathcal{M}'$ is base-orderable if and only if $\mathcal{M}$ is base-orderable.*

ALGORITHM 7: Iterated Swaps

Input: Constraints based on a base-orderable matroid $\mathcal{M}$; n agents with additive valuations; A set M of items with $|M| = n \cdot \text{rank}(\mathcal{M})$.

1: **Initialize**: $\mathbf{X} \leftarrow$ a complete feasible SWM allocation (by Theorem 9).
2: **while** $\mathbf{X}$ is not EF1 **do**
3: Find $i, j \in N$ such that agent i envies agent j by more than one item.
4: Find a feasible exchange bijection $\mu : X_i \to X_j$.
5: Find an item $g_i \in X_i$ such that $v_i(\mu(g_i)) > v_i(g_i)$.
6: Swap items g_i and $\mu(g_i)$.
7: **end while**
8: **return** the allocation $\mathbf{X}$.

By the above observation, one can assume, without loss of generality, that there are exactly $n \cdot r$ items, and consider only allocations in which each agent receives a basis of $\mathcal{M}$. We can now present the Iterated Swaps scheme, presented in Algorithm 7.

The algorithm starts by finding a feasible social welfare maximizing (SWM) allocation $\mathbf{X}$, using Theorem 9. As long as there exist agents i, j that violate the EF1 condition, the algorithm swaps a pair of items between i and j, such that the utility of i (the envious agent) increases. In the next subsections we will present two settings in which Iterated Swaps is indeed guaranteed to terminate in polynomial time with an EF1 allocation.

7.3 Three Agents with Binary Valuations

Below, we show that Iterated Swaps finds an EF1 allocation for $n = 3$ agents with heterogeneous *binary* valuations.

Theorem 10. *Consider an instance with identical constraints based on a base-orderable matroid $\mathcal{M}$. For every three agents with heterogeneous binary valuations, the Iterated Swaps algorithm (Algorithm 7) finds an EF1 and SWM allocation.*

The algorithm uses a polynomial number of arithmetic operations and calls to an independence oracle of $\mathcal{M}$.

To prove Theorem 10 we use the following lemma.

Lemma 9. *Consider a setting with binary valuations. Let $\mathbf{X}$ be a SWM allocation in which agent i envies agent j. Let $\mu : X_i \to X_j$ be a feasible-exchange bijection. Then there is an item $g_i \in X_i$ for which $v_i(g_i) = v_j(g_i) = 0$ and $v_i(g_j) = v_j(g_j) = 1$, where $g_j = \mu(g_i)$.*

Proof. By additivity,

$$v_i(X_j) = \sum_{g \in X_j} v_i(g) = \sum_{g \in X_i} v_i(\mu(g));$$
$$v_i(X_i) = \sum_{g \in X_i} v_i(g).$$

Since i envies j, $v_i(X_j) > v_i(X_i)$, so at least one term in the first sum must be larger than the corresponding term in the second sum. So there is some $g_i \in X_i$ for which:

$$v_i(\mu(g_i)) > v_i(g_i).$$

Let $g_j := \mu(g_i)$. Since valuations are binary, $v_i(g_j) > v_i(g_i)$ implies $v_i(g_j) = 1$ and $v_i(g_i) = 0$. Since μ is a feasible-exchange bijection, swapping g_i and g_j yields a feasible allocation. Since $\mathbf{X}$ maximizes the sum of utilities among all feasible allocations, the swap cannot increase the sum of utilities; therefore, we must have $v_j(g_j) = 1$ and $v_j(g_i) = 0$ too. □

We call the exchange described by Lemma 9 a *smart swap.*

Lemma 10. *Let* $\mathbf{X}$ *be a SWM allocation where* $\textsc{Envy}^+_{\mathbf{X}}(i,j) > 1$*, and let* $\mathbf{X}'$ *be an allocation obtained from* $\mathbf{X}$ *by a smart swap between agents* i *and* j*. Then:*
(a) $\mathbf{X}'$ *is a SWM allocation.*
(b) $\textsc{Envy}^+_{\mathbf{X}'}(i,j) = \textsc{Envy}^+_{\mathbf{X}}(i,j) - 2$*;*
(c) $v_i(X'_j) \geq v_i(X'_i)$*;*
(d) $\textsc{Envy}^+_{\mathbf{X}'}(j,i) = 0$.

Proof. (a) The smart swap decreases the utility of j by $v_j(g_j) - v_j(g_i) = 1$, and increases the utility of i by $v_i(g_j) - v_i(g_i) = 1$, and does not change the utilities of other agents. So the total sum of utilities does not change.

(b) After the smart swap, $v_i(X'_j) = v_i(X_j) - v_i(g_j) = v_i(X_j) - 1$, and $v_i(X'_i) = v_i(X_i) + v_i(g_j) = v_i(X_i) + 1$. It follows that $\textsc{Envy}^+_{\mathbf{X}'}(i,j) = v_i(X'_j) - v_i(X'_i) = v_i(X_j) - 1 - (v_i(X_i) + 1) = \textsc{Envy}^+_{\mathbf{X}}(i,j) - 2$.

(c) Before the swap, we had $\textsc{Envy}^+_{\mathbf{X}}(i,j) \geq 2$, so $v_i(X_j) \geq v_i(X_i) + 2$. The swap decreased the difference in utilities by 2. Therefore, after the swap, we still have $v_i(X'_j) \geq v_i(X'_i)$.

(d) If we had $\textsc{Envy}^+_{\mathbf{X}'}(j,i) > 0$, then giving X'_i to j and X'_j to i would increase the utility of j and not decrease the utility of i (by 3), contradicting SWM. □

We are now ready to prove Theorem 10. In the proof, when we mention a change in $\textsc{Envy}^+_{\mathbf{X}}(i,j)$ we refer to the change in the positive envy of agent i to agent j between allocations $\mathbf{X}$ and $\mathbf{X}'$; i.e., to $\textsc{Envy}^+_{\mathbf{X}'}(i,j) - \textsc{Envy}^+_{\mathbf{X}}(i,j)$.

Proof of Theorem 10. Let $\mathbf{X}$ be a complete feasible SWM allocation. Let $\Phi(\cdot)$ be the following potential function:

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

If $\mathbf{X}$ is EF1, we are done. Otherwise, by Lemmas 9 and 10, there must exist a smart swap between i, j such that the social welfare remains unchanged, $\textsc{Envy}^+_{\mathbf{X}}(i,j)$ drops by 2 and $\textsc{Envy}^+_{\mathbf{X}}(j,i)$ remains 0. Thus, $\textsc{Envy}^+_{\mathbf{X}}(i,j) + \textsc{Envy}^+_{\mathbf{X}}(j,i)$ drops by 2. Let us next consider the positive envy that might be added due to terms of Φ that include the third agent, denote it by k.

1. $\textsc{Envy}^+_{\mathbf{X}}(i,k)$ cannot increase, as the smart swap increases i's utility, while $v_i(X_k)$ does not change.

2. $\textsc{Envy}^+_{\mathbf{X}}(k,i)$ increases by at most 1: the largest possible increase in $v_k(X'_i)$ is 1, while $v_k(X_k)$ does not change.

3. $\text{Envy}^+_{\mathbf{X}}(k, j)$ increases by at most 1: the largest possible increase in $v_k(X'_j)$ is 1, while $v_k(X_k)$ does not change.

4. $\text{Envy}^+_{\mathbf{X}}(j, k)$ increases by at most 1, as this is the exact decrease in $v_j(X'_j)$, while $v_j(X_k)$ does not change.

We next claim that among the terms that may increase by 1 (that is, terms $2, 3$ and 4), no two of them can increase simultaneously:

- $\text{Envy}^+_{\mathbf{X}}(k, j), \text{Envy}^+_{\mathbf{X}}(j, k)$ cannot increase simultaneously as this would create an envy-cycle, contradicting SWM.

- $\text{Envy}^+_{\mathbf{X}}(k, i), \text{Envy}^+_{\mathbf{X}}(j, k)$ cannot increase simultaneously, as this together with the fact that $v_i(X'_j) \geq v_i(X'_i)$ contradicts SWM: shifting bundles along the cycle $i \to j \to k \to i$ strictly increases the sum of utilities.

- $\text{Envy}^+_{\mathbf{X}}(k, i), \text{Envy}^+_{\mathbf{X}}(k, j)$ cannot increase simultaneously as the sum of k's valuations to i's and j's bundles is fixed, that is, $v_k(X_i) + v_k(X_j) = v_k(X'_i) + v_k(X'_j)$.

We conclude that in every iteration the potential function drops by at least 1. Indeed, $\text{Envy}^+_{\mathbf{X}}(i, j)$ drops by 2, $\text{Envy}^+_{\mathbf{X}}(j, i)$ remains 0, $\text{Envy}^+_{\mathbf{X}}(i, k)$ does not change, and among $\text{Envy}^+_{\mathbf{X}}(k, i)$, $\text{Envy}^+_{\mathbf{X}}(k, j)$, $\text{Envy}^+_{\mathbf{X}}(j, k)$ only one can increase, by at most 1.

As the valuations are binary, the initial value of the potential function Φ is bounded by $|M| = m$. At every step it drops by at least one, so the algorithm stops after at most m iterations.

By Lemma 8, line 4 can be executed in polynomial time. All other lines can clearly be executed in polynomial time too. Therefore, the run-time is polynomial. □

7.4 Two Agents with Additive Valuations

The case of three agents with heterogeneous additive valuations remains open. Below, we show that for *two* agents with heterogeneous additive valuations, an EF1 allocation always exists. Suppose the agents' valuations are v_1 and v_2. Using the cut-and-choose algorithm, we can reduce the problem to the case of *identical* valuations:

- Find an allocation that is EF1 for two agents with identical valuation v_1.

- Let agent 2 pick a favorite bundle (the allocation is envy-free for 2).

- Give the other bundle to agent 1 (the allocation is EF1 for 1).

It remains to show how to find an EF1 allocation for agents with identical valuations. Biswas and Barman (2018) [Section 7 in the full version] presented an algorithm that finds an EF1 allocation for n agents with identical valuations, with constraints based on a *laminar* matroid — a special case of a BO matroid. In fact, their algorithm can be both simplified and extended to BO matroids using our pre-processing step and the Iterated Swaps scheme.

Algorithm 7 should be adapted in the following way:

- In line 3, the envious agent i is chosen such that his current value is smallest:

$$i \in \operatorname*{argmin}_{i \in N} v(X_i).$$

  Note that, since both the valuations and the constraints are identical, if some agent is envious, then any agent with the smallest value is envious too, so this choice is possible.

- In line 5, the pair of items to swap is chosen such that the value-difference is largest:

$$g_i \in \operatorname*{argmax}_{g_i \in X_i} v(\mu(g_i)) - v(g_i).$$

With these two choices, it can be proved that the algorithm terminates after polynomially many iterations. The proof is similar to Theorem 4 of Biswas and Barman (2018) [pages 16–18]; for completeness, we present a stand-alone proof below. The proof is based on several claims.

Claim 1. *The smallest bundle value,* $\min_{k \in N} v(X_k)$*, does not decrease during the algorithm.*

Proof. The only agent whose value decreases in each iteration is the agent j, who is envied by i. Since the envy is by more than one item, j's value after the swap is still larger than i's value before the swap. Therefore, the globally smallest value does not decrease. □

Claim 2. *If some agent i is picked as the envious agent in some iteration, then i is never envied by more than one item in any later iteration (and therefore is not picked as the envied agent j).*

Proof. The proof is by induction on the algorithm iterations.

The basis is the first iteration in which agent i is chosen as the envious agent; denote this iteration by t_i. i's value starts at $\min_{i \in N} v(X_i)$ and increases by at most $v(\mu(g_i))$. Therefore, if $\mu(g_i)$ is removed from X_i, then the remaining value in X_i is at most $\min_{i \in N} v(X_i)$. Therefore, i is not envied by more than one item in the next iteration.

Suppose the claim holds for iterations $t_i, t_i + 1, \ldots, t_i + k$ for some $k \geq 0$. If, in iteration $t_i + k$, agent i is again chosen as the envious agent, then the induction basis applies again, and i is not envied by more than one item in the next iteration. Otherwise, by the induction assumption, agent i is not chosen as the envied agent, so his bundle X_i does not change. By Claim 1, the smallest agent value does not decrease; therefore, i is still not envied by more than one item in the next iteration. □

Claim 3. *Consider a setting with identical additive valuations v. Let $\mathbf{X}$ be an allocation in which agent i envies agent j by more than one item. For any bijection $\mu : X_i \to X_j$:*

$$\max_{g_i \in X_i} [v(\mu(g_i)) - v(g_i)] > v(X_j)/m^2.$$

Proof. Since the envy is by more than one item, the difference $v(X_j) - v(X_i)$ is larger than $\max_{g_j \in X_j} v(g_j)$, which is at least as large as $v(X_j)/|X_j|$.

Since $v(X_j) - v(X_i) = \sum_{g_i \in X_i} (v(\mu(g_i)) - v(g_i))$, there exists at least one g_i for which $v(\mu(g_i)) - v(g_i) \geq [v(X_j) - v(X_i)]/|X_i|$.

Combining these inequalities gives:

$$\begin{aligned}\max_{g_i \in X_i} v(\mu(g_i)) - v(g_i) &> (v(X_j)/|X_j|)/|X_i| \\ &> v(X_j)/m^2.\end{aligned}$$ □

The next claim is a general claim about sequences of subsets of numbers.

Claim 4. *Let $r > 1$ be a real number. Let M be a multiset of m positive real numbers. Let $\mathbf{Y} = Y_1, Y_2, \ldots$ be a sequence of subsets of M, such that*

$$\sum_{x \in Y_{t+1}} x \;>\; r \cdot \sum_{x \in Y_t} x \qquad \textit{for all } t \geq 1,$$

that is, the sums of subsets in the sequence increase by a factor larger than r at each step. Then the length of the sequence $\mathbf{Y}$ is at most $R \cdot (m+1)$, where $R = \lceil 1/\log_2 r \rceil$.[9]

Proof. Define a sub-sequence $\mathbf{Z}$ of $\mathbf{Y}$, where $Z_t := Y_{R \cdot t}$ for all $t \geq 1$. The condition on the sequence $\mathbf{Y}$ implies that

$$\sum_{x \in Z_{t+1}} x \;>\; r^R \cdot \sum_{x \in Z_t} x \;\geq\; 2 \cdot \sum_{x \in Z_t} x \qquad \text{for all } t \geq 1,$$

that is, the sums of subsets in the sequence more than double at each step. This implies that:

$$\sum_{x \in Z_{t+1}} x \;>\; \sum_{x \in Z_1} x + \cdots + \sum_{x \in Z_t} x \qquad \text{for all } t \geq 1,$$

so each subset Z_{t+1} must contain a new element of M, that is not contained in any of $Z_1, \ldots, Z_t$. Therefore, the length of $\mathbf{Z}$ is at most $m+1$, so the length of $\mathbf{Y}$ is at most $R \cdot (m+1)$. □

Theorem 11. *In any setting with identical additive valuations and identical BO matroid constraints, the Iterated Swaps algorithm with the two choices mentioned above (choosing $i \in \arg\min_N v(X_i)$ and $g_i \in \arg\max_{X_i} v(\mu(g_i)) - v(g_i)$) finds an EF1 allocation in $O(nm^3)$ iterations.*

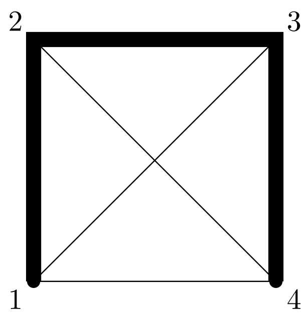
Proof. Claim 3 implies that, at each iteration, the envious agent i gains a value of at least $v(X_j)/m^2$, which is larger than $v(X_i)/m^2$. Therefore, i's value is multiplied by a factor larger than $r := (1 + 1/m^2)$. By Claim 2, i's value never decreases during the following iterations. Applying Claim 4 to the sequence of iterations in which i is selected as the envious agent implies that i can be envious at most $R \cdot (m+1)$ times, where $R := \lceil 1/\log_2 r \rceil$. Note that $r^{m^2} > 2$ for all $m \geq 2$, so $1/\log_2 r < m^2$, so $R \leq m^2$. Therefore, the algorithm must terminate after at most $Rn(m+1) = O(nm^3)$ iterations. □

9. Note that $R > 1$ when $r < 2$ and $R = 1$ when $r \geq 2$.

7.5 Non Base-Orderable Matroids

The Iterated Swaps technique may fail for matroids that are not base-orderable, even for two agents with identical binary valuations. The "weak link" is Lemma 9, as the following example shows.

Example 2. Consider the *graphical matroid on* K_4 — the clique on four vertices. Denote the vertices of K_4 by $1, 2, 3, 4$ and its edges by $E = \{12, 13, 14, 23, 24, 34\}$. The K_4 graphical matroid is a matroid over the ground-set E, whose independent sets are the trees in K_4. Consider the two bases $\{12, 23, 34\}$ (thick) and $\{24, 41, 13\}$ (thin):



The only feasible swap for 12 is with 14, and similarly the only feasible swap for 34 is with 14, so there is no feasible-exchange bijection. This shows that the graphical matroid on K_4 is not base-orderable.

Suppose now that the agents have identical valuations, as in the following table:

Element (edge of K_4)	Value
12	0
23	1
34	0
13	1
14	0
24	1

Note that, with identical valuations, all allocations are SWM. Consider the allocation in which Alice holds the first three elements and Bob holds the last three elements. Then Alice envies Bob, but there is no swap that increases Alice's utility while keeping the bundles of both agents feasible. Moreover, suppose there are two copies of K_4, and in both copies the allocation is the same as above. Then, Alice envies Bob by two items, but there is no feasible single-item swap that can reduce her envy. Thus, although an EF1 allocation exists, it might not be attainable by single-item swaps from an arbitrary SWM allocation. □

We have a positive result for the special case of *unary valuations*, in which each agent values every item at 1 (note that the items are still different due to the constraints).

Theorem 12. *Consider an instance with any number of agents, with heterogeneous matroid constraints and with unary valuations. If there exists a complete feasible allocation (that is, Assumption 1 holds), then there exists a complete feasible F-EF1 allocation.*

Proof. Benabbou et al. (2021) prove that, for agents whose valuations are matroid-rank functions, there always exists an EF1 allocation that maximizes social welfare. When agents have unary valuations, the social welfare of every complete feasible allocation is exactly m (the number of items). Since a complete feasible allocation exists, the maximum attainable social welfare is m. The agents' feasible-valuation functions $\hat{v}_i$ are matroid-rank functions. Therefore, by Benabbou et al. (2021), there exists an F-EF1 allocation with social welfare m. This allocation must be a complete feasible allocation. □

8. Impossibility Results

In this section we give some intuition for why previous approaches fail in the case of heterogeneous constraints, and provide impossibility results for settings beyond the ones considered in this paper. All examples in this section involve two agents.

8.1 Partition Matroids, Maximum Nash Welfare

The following example shows that a maximum Nash welfare (MNW) outcome may not be F-EF1 in settings with feasibility constraints, even under identical partition matroid constraints and binary valuations. We note that the existence of such an example has been mentioned by Biswas and Barman (2018) without a proof; we include it here for completeness.

Example 3. Consider the setting and allocation illustrated in Table 2. There are two agents and two categories. Category 1 has 4 items and capacity 2 for both agents; Category 2 has 6 items and capacity 3 for both agents. From the items in C^2, Alice gets value 0 and Bob gets value 3 in any feasible allocation. From the items in C^1, Alice and Bob together get a total value of 2; if Alice gets 0 and Bob gets 2, then the Nash welfare is 0; if Alice gets 1 and Bob gets 1, then the Nash welfare is $1 \cdot (1 + 3) = 4$; if Alice gets 2 and Bob gets 0, then the Nash welfare is $2 \cdot (0 + 3) = 6$. Therefore, the maximum Nash welfare is 6, attained by the allocation marked by asterisks in Table 2. This allocation is not EF1, since $v_B(X_A) = 5$ so Bob envies Alice by more than one item (note that F-EF1 and EF1 are equivalent in this case, since both agents have the same constraints).

Category	C^1		C^2	
Items	w_1, w_2	x_1, x_2	$y_1 \ldots, y_3$	$z_1 \ldots, z_3$
v_A	1 *	0	0 *	0
v_B	1	0 *	1	1 *

Table 2: An example of agents with identical partition matroid constraints and binary valuations where MNW does not imply EF1.

Note that Benabbou et al. (2021) prove that MNW always implies EF1 for submodular valuations with binary marginals. However, they consider *clean* allocations, where items with 0 marginal value are not allocated. In contrast, we consider *complete* allocations, in which all items must be allocated. Indeed, if we "clean" the allocation in Table 2 by having Alice dispose of the three items she does not desire in category C^2, the allocation becomes EF1 while remaining MNW. The same reasoning (and same example) applies also to the *prioritized egalitarian* mechanism introduced by Babaioff et al. (2021). Specifically, this mechanism gives a *clean* ("non-redundant" in their terminology) Lorentz-dominating allocation, which is shown to be EF1 (and even EFX, cf. Subsection 8.4). However, the obtained allocation is not complete.

8.2 Partition Matroids, Heterogeneous Categories

The following example shows that if agents have partition matroid constraints, where the partitions of the items into categories is not the same, then an F-EF1 allocation might not exist, even if the valuations are identical and binary.

Example 4 (Heterogeneous categories, no F-EF1)**.** Consider a setting with four items and two agents with identical binary valuations:

Items	x_1, x_2	y_1, y_2
$v_A = v_B$	1	0

Suppose that

- Alice's partition has two categories, $C_A^1 = \{x_1, y_1\}$ and $C_A^2 = \{x_2, y_2\}$, both with capacity 1;
- Bob's partition has three categories: $C_B^1 = \{x_1\}$ with capacity 1, $C_B^2 = \{x_2\}$ with capacity 1, and $C_B^3 = \{y_1, y_2\}$ with capacity 0.

There is a unique feasible allocation, in which Bob gets x_1, x_2 and Alice gets y_1, y_2. Bob's bundle is feasible for Alice, and Alice envies Bob by more than one item.

Remark 1. Example 4 has two items worth 0 to both agents. What happens if we do not allow such worthless items?

If we replace these items with two items worth ϵ to both agents, for some $\epsilon \in (0, 1/2)$, then the same arguments show that an F-EF1 allocation does not exist (we used 0 to show impossibility for *binary* valuations).

On the other hand, if we allow only items that are worth 1 to all agents (that is, we require *unary* valuations), then an F-EF1 allocation always exists by Theorem 12.

Remark 2. Example 4 can be modified to avoid categories with zero capacities:

Items	$x_1, \ldots, x_4$	$y_1, \ldots, y_4$
$v_A = v_B$	1	0

- Alice's partition has four categories, $C_A^j = \{x_j, y_j\}$ for $j \in [4]$, with capacity 1 each;
- Bob's partition has five categories: $C_B^j = \{x_j\}$ for $j \in [4]$ with capacity 1 each, and $C_B^5 = \{y_1, \ldots, y_4\}$ with capacity 1.

In every complete feasible allocation, Bob gets at most one y_j item, so Alice must get at least three y_j items, so she can get at most one x_j item, so Bob must get at least three x_j items. Now, $\hat{v}_A(X_A) \leq 1$ while $\hat{v}_A(X_B) \geq 3$ (since the subset of X_B containing the three x_j items is feasible for Alice). Hence, Alice F-envies Bob by more than one item.

8.3 Non-Existence of EF1 for Non-Matroid Constraints

In this subsection we consider natural classes of set-systems that are not matroidal. We show that, when the constraints are based on such set-systems, a complete EF1 allocation might not exist, even when all agents have identical constraints and identical binary valuations.

Definition 17. Let $\mathcal{I} \subseteq 2^M$ be a set-system (a set of subsets) of M. Two elements $x, y \in M$ are called *complementary for* $\mathcal{I}$ if, for every partition of M into two subsets $X_1, X_2 \in \mathcal{I}$ (with $X_1 \cap X_2 = \emptyset$ and $X_1 \cup X_2 = M$), either $\{x, y\} \subseteq X_1$ or $\{x, y\} \subseteq X_2$.

Example 5. Let $M_1 = \{1, 2, 3\}$ and $\mathcal{I}_1 = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$. There is only one partition of M into two subsets in $\mathcal{I}_1$, namely, $\{2\}$ and $\{1, 3\}$. This shows that 1 and 3 are complementary elements.

As another example, let $M_2 = \{1, 2, 3, 4\}$ and $\mathcal{I}_2 = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 3\}, \{2, 4\}\}$. Here, 1 and 3 are complementary elements, and so are 2 and 4.

Proposition 1. *Suppose there are two agents with identical constraints represented by some $\mathcal{I} \subseteq 2^M$. If there are complementary items for $\mathcal{I}$, then there exists a binary valuation function $v : M \to \{0, 1\}$ such that, when $v_1 = v_2 = v$, there is no complete feasible EF1 allocation.*

Proof. Define v such that both agents value both complementary items at 1 and the other items at 0. In every feasible allocation, one agent gets none of the complementary items, and thus envies the other agent who gets both these items, and the envy is by two items. $\square$

There are several natural constraints with complementary items.

Example 6 (Matroid-intersection constraints, matching constraints, conflict-graph constraints)**.** Consider the allocation of course seats among students, where the seats are categorized both by their time and by their subject, and a student should get at most a single seat per time and at most a single seat per subject. Suppose there are two subjects, physics and chemistry; each of them is given in two time-slots, morning and evening. The items (physics,morning) and (chemistry,evening) are complementary, since in the (unique) partition of the four seats into two feasible subsets, these two seats appear together. In fact, if we set 1 = (physics,morning), 2 = (chemistry,morning), 3 = (chemistry,evening), 4 = (physics,evening), then the set of feasible subsets is exactly the set $\mathcal{I}_2$ from Example 5. By Proposition 1, an EF1 allocation may not exist.

In general, the above constraints can be represented by an intersection of two partition matroids: one matroid $(M, \mathcal{I}_t)$ partitions the seats into two categories by their time, and another matroid $(M, \mathcal{I}_s)$ partitions them by their subject, and the capacities of all categories are 1. The feasible bundles are the bundles in $\mathcal{I}_t \cap \mathcal{I}_s$.

The above constraints can also be represented by the set of *matchings* in the bipartite graph in Figure 1, where the items are the edges. Matching constraints in bipartite graphs can be formed as an intersection of two partition matroids: one partitions the edges into categories based on their leftmost endpoint, and the other partitions the edges into categories based on their rightmost endpoints, and all categories in both matroids have a capacity of 1.

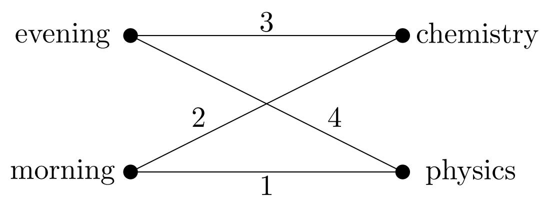


Figure 1: Matroid intersection graph

Figure 2: Conflict graph

The same constraints can be represented as *conflict-graph constraints*, recently studied by Hummel and Hetland (2022a). Suppose the items are the vertices of the graph shown in Figure 2. Edges denote conflicts, and the feasible sets are the set of non-adjacent vertices. Two diagonally-opposite vertices are complementary, so by Proposition 1 an EF1 allocation may not exist.

Example 7 (Budget constraints)**.** Budget constraints were recently studied by Wu et al. (2021), Gan et al. (2021). Suppose there is one item 2 with a cost of 20, and two items $1, 3$ with a cost of 10 each. There are two agents with budget 20. The set of feasible subsets is exactly the set $\mathcal{I}_1$ from Example 5. The only complete feasible partition is to $\{2\}$ and $\{1, 3\}$. The items $1, 3$ are complementary, so by Proposition 1 an EF1 allocation might not exist.

Note that the same set $\mathcal{I}_1$ can be represented as an intersection of two partition matroids (one with categories $\{1, 2\}$ and $\{3\}$ and one with categories $\{1\}$ and $\{2, 3\}$, both with capacity 1), or as the set of matchings in a graph (the graph of Figure 1 with edge 4 removed), or using a conflict graph (the graph of Figure 2 with vertex 4 removed).

Remark 3. If $(M, \mathcal{I})$ is a matroid, and there is at least one partition of M into two independent sets, then there are no complementary items for $\mathcal{I}$. This follows from the symmetric basis exchange property (Brualdi, 1969).

The opposite is not necessarily true. For example, suppose the elements of M are arranged on a line and $\mathcal{I}$ contains all the connected subsets along the line. This constraint is not a matroid, since it is not downward-closed. But it has no complementary items. Indeed, an EF1 allocation exists for any number of agents with binary valuations (Bilò et al., 2018).

As another example, consider a budget constraint with a budget of 7, and suppose there are four items with costs $1, 2, 3, 4$. This constraint is downward-closed, but it is not a matroid, since there are maximal feasible sets of different cardinalities ($1, 2, 4$ and $3, 4$). But it has no complementary items: 1 and 2 are separated by the feasible partition $\{1,3\},\{2,4\}$; 3 is separated from the other items by the feasible partition $\{3\},\{1,2,4\}$; and 4 is separated from the other items by the feasible partition $\{4\},\{1,2,3\}$.

Therefore, characterizing the constraints for which an EF1 allocation is guaranteed to exist remains an open problem.

8.4 Non-Existence of EFX, Uniform Matroids

An *envy-free up to any good* (EFX) allocation is a feasible allocation $\mathbf{X}$ where for every pair of agents i, j, for every good g in j's bundle, $v_i(X_i) \geq v_i(X_j \setminus \{g\})$. Clearly, EFX is stronger than EF1. EFX has been recently shown to exist in the unconstrained settings for up to 3 agents with additive valuations (Chaudhury, Garg, & Mehlhorn, 2020). However, under constrained settings an EFX allocation may not exist even in the simple setting of two agents with identical uniform matroid constraints and identical binary valuations.

Example 8. There are four items and two agents with an identical valuation, $v(x_1) = v(x_2) = v(x_3) = 0$ and $v(x_4) = 1$. The capacity of both agents is 2. In every complete feasible allocation (i.e., allocating 2 items to each agent), the agent who does not get x_4 is envious beyond EFX.

9. Conclusions and Future Directions

We have identified several sub-domains of the fair allocation problem, in which a complete and feasible EF1 allocation is guaranteed to exist. Each sub-domain represents a limitation on either the number of agents, or the type of allowed valuations, or the type of allowed constraints. At the same time, we have identified several sub-domains in which a complete and feasible EF1 allocation might *not* exist. Our results do not provide a complete characterization: some sub-domains are covered neither by our positive results nor by our negative results. This suggests several open questions.

1. Consider a setting with n agents with additive heterogeneous valuations, partition matroids with heterogeneous capacities, and *three* or more categories. Does an EF1 allocation exist? (Section 3 handles at most two categories).

2. Consider a setting with n agents with additive identical valuations. Is there a class of matroids, besides partition matroids with the same categories (Section 4), for which an EF1 allocation exists even when the constraints are heterogeneous?

3. Consider a setting with n agents with binary valuations, partition matroids with heterogeneous capacities, and the capacities may be *two* or more. Does an EF1 and Pareto-efficient allocation exist? (Section 5.2 handles capacities in $\{0, 1\}$).

4. Consider a setting with *three* or more agents with heterogeneous additive valuations and partition matroids with heterogeneous capacities. Does an EF1 allocation exist? (Section 6 handles two agents).

5. Consider a setting with *four* or more agents with binary valuations and BO matroid constraints, or even *three* agents with binary valuations and general matroid constraints, or three agents with *additive* valuations and BO matroid constraints. Does

an EF1 allocation exist? (Section 7 requires three agents, binary valuations, and BO matroids).

Another interesting direction is extending our results to allocation of *chores* (items with negative utilities) in addition to goods.

Acknowledgments

A preliminary version appeared in the proceedings of AAAI 2021 (Dror, Feldman, & Segal-Halevi, 2021), without most of the proofs. This version contains all omitted proofs, a more uptodate literature survey, a simpler proof of Theorem 5, simpler algorithms and proofs in Section 7, stronger version of Theorem 9 and more general non-existence results in Section 8.3.

Amitay Dror and Michal Feldman received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 866132), and the Israel Science Foundation (grant number 317/17). Erel Segal-Halevi received funding from the Israel Science Foundation (grant number 712/20).

We are grateful to Jonathan Turner, Jan Vondrak, Chandra Chekuri, Tony Huynh, Yuval Filmus, Kevin Long, Enrico Gregorio, Siddharth Barman, Arpita Biswas, and anonymous reviewers of AAAI 2021 and JAIR for their invaluable comments.

Appendix A. Alternative Definitions of Feasible Fairness

Recall the definition of F-EF1 from Section 2.2:

> An allocation $\mathbf{X}$ is *F-EF1* iff for every $i, j \in N$, there exists a subset $Z' \subseteq X_j$ with $|Z'| \leq 1$, such that $v_i(X_i) \geq \hat{v}_i(X_j \setminus Z')$.

This definition compares agent i's bundle to X_j after first removing an item from X_j, and then considering the most valuable feasible subset within the remainder. The condition can be written as follows:

$$v_i(X_i) \geq \min_{Z' \subseteq X_j, |Z'| \leq 1} \max_{Y'_j \in \text{FEAS}_i(X_j \setminus Z')} v_i(Y'_j).$$

Alternatively, we could first consider a feasible subset of X_j, and then remove an item. This yields the following definition:

Definition 18 (weak F-EF1)**.** An allocation $\mathbf{X}$ is *weak F-EF1* iff for every $i, j \in N$, for any $Y_j \in \text{FEAS}_i(X_j)$, there exists a subset $Z \subseteq Y_j$ with $|Z| \leq 1$, such that $v_i(X_i) \geq v_i(Y_j \setminus Z)$. In other words:

$$v_i(X_i) \geq \max_{Y_j \in \text{FEAS}_i(X_j)} \min_{Z \subseteq Y_j, |Z| \leq 1} v_i(Y_j \setminus Z).$$

The definition makes sense for downward-closed constraints, since then the fact that Y_j is feasible implies that $Y_j \setminus Z$ is feasible too. The following lemma justifies the adjective "weak".

Lemma 11. *For downward-closed constraints,*

(a) every F-EF1 allocation is weak F-EF1.

(b) Not every weak F-EF1 allocation is F-EF1.

Proof. (a) If $\mathbf{X}$ is F-EF1, there exists $Z' \subseteq X_j$ with $|Z'| \leq 1$, such that $v_i(X_i) \geq \hat{v}_i(X_j \setminus Z')$.

Equivalently, $v_i(X_i) \geq v_i(Y'_j)$ for any $Y'_j \in \textsc{feas}_i(X_j \setminus Z')$.

Let Y_j be any set in $\textsc{feas}_i(X_j)$. Let $Z := Z' \cap Y_j$. Note that $Z \subseteq Y_j$ and $|Z| \leq 1$. Moreover, $Y_j \setminus Z \subseteq X_j \setminus Z'$ since $Y_j \subseteq X_j$. Moreover, $Y_j \setminus Z$ is feasible for i, since Y_j is feasible for i and the constraints are downward-closed. Therefore, $Y_j \setminus Z \in \textsc{feas}_i(X_j \setminus Z')$, so $v_i(X_i) \geq v_i(Y_j \setminus Z)$, satisfying the requirement for weak F-EF1.

(b) Consider a uniform matroid and two agents, with two items worth 1 to both agents, and let $k_A = 1, k_B = 2$. Let $\mathbf{X}$ be the allocation giving both items to Bob. This allocation is weak F-EF1: every subset $Y_B \in \textsc{feas}_1(X_B)$ contains a single object, and once this object is removed, Alice does not envy.

However, $\mathbf{X}$ is not F-EF1. If any item is removed from Bob's bundle, then the remaining bundle contains a single item, it is feasible for Alice, and Alice values it more than her own (empty) bundle. Therefore, Alice F-envies Bob.

□

All algorithms in this paper return F-EF1 allocations, thus weak F-EF1 as well.

Conversely, all impossibility results in this paper (see Section 8) continue to hold also with respect to weak F-EF1 too: this is obvious in Section 8.1, Section 8.3 and Section 8.4 since they use identical constraints. With identical constraints, all bundles in a feasible allocation are feasible for all agents, so all three fairness notions (EF1, F-EF1 and weak F-EF1) coincide.

In Section 8.2 (which uses heterogeneous partition-matroid constraints), there is a unique feasible allocation, in which Bob gets two items worth 1 for both agents, and Alice gets two items worth 0 for both agents. Bob's bundle is feasible for Alice, and she envies Bob even after one item is removed from it, so the allocation is not weak F-EF1.

An even weaker definition of feasible fairness presumes that an agent can only envy a feasible bundle. That is, if the bundle of agent j is not feasible for agent i, then i does not envy j at all. This leads to the following variant of EF1:

Definition 19 (very-weak F-EF1)**.** An allocation $\mathbf{X}$ is *very-weak F-EF1* iff for every $i, j \in N$, either X_j is not feasible for agent i, or there exists a subset $Z \subseteq X_j$ with $|Z| \leq 1$, such that $v_i(X_i) \geq v_i(X_j \setminus Z)$.

Clearly, every weak F-EF1 allocation is very-weak F-EF1, so all our positive results are valid for very-weak F-EF1. Our negative results, too, are valid for very-weak F-EF1. The reasons are exactly the same as for weak F-EF1: in Sections 8.1, 8.3 and 8.4 the constraints are identical so all variants of EF1 are equivalent; in Section 8.2, in the unique feasible allocation, Bob's bundle is feasible for Alice, and she envies Bob by more than one item, so the allocation is not very-weak F-EF1.

Appendix B. Appendix for Section 7

Lemma 12. *Let $\mathcal{M} = (M, \mathcal{I})$ be a matroid, $\mathcal{M}' = (M', \mathcal{I}')$ be its free extension with new item x^{new}. Then $\mathcal{M}'$ is base-orderable if and only if $\mathcal{M}$ is base-orderable.*

Proof. If $\mathcal{M}'$ is base-orderable, then every two bases of $\mathcal{M}'$ have a feasible-exchange bijection. Since all bases of $\mathcal{M}$ are bases of $\mathcal{M}'$, the same holds for $\mathcal{M}$ too.

Conversely, suppose $\mathcal{M}$ is base-orderable, and let $I', J' \in \mathcal{I}'$ be two bases of $\mathcal{M}'$. We consider several cases.

Case 1: Both I' and J' do not contain x^{new}. Then both are bases of $\mathcal{M}$, so they have a feasible-exchange bijection.

Case 2: I' contains x^{new} while J' does not. So $I' = I + x^{\text{new}}$, where $I \in \mathcal{I}$, and J' is a basis of $\mathcal{M}$. Let $I + y$ be any basis of $\mathcal{M}$ that contains I, where $y \in M$. Since $\mathcal{M}$ is BO, there is a feasible-exchange bijection $\mu : I + y \to J'$. Define a bijection $\mu' : I + x^{\text{new}} \to J'$ by:

$$\begin{aligned} \mu'(x) &= \mu(x) \quad \text{for } x \in I; \\ \mu'(x^{\text{new}}) &= \mu(y). \end{aligned}$$

We now show that μ' is a feasible-exchange bijection.

- For all $x \in I$, we have

  $$(I + x^{\text{new}}) - x + \mu'(x) = (I - x + \mu(x)) + x^{\text{new}}.$$

  Since μ is a feasible-exchange bijection, $(I+y)-x+\mu(x) \in \mathcal{I}$. By downward-closedness, $I - x + \mu(x) \in \mathcal{I}$. By definition of the free extension, $(I - x + \mu(x)) + x^{\text{new}} \in \mathcal{I}'$. Additionally,

  $$J' - \mu'(x) + x = J' - \mu(x) + x.$$

  Since μ is a feasible-exchange bijection, the latter set is in $\mathcal{I}$, which is contained in $\mathcal{I}'$.

- For $x = x^{\text{new}}$, we have

  $$(I + x^{\text{new}}) - x^{\text{new}} + \mu'(x^{\text{new}}) = I + \mu(y) = (I + y) - y + \mu(y).$$

  Since μ is a feasible-exchange bijection, the latter set is in $\mathcal{I}$, which is contained in $\mathcal{I}'$. Additionally,

  $$J' - \mu'(x^{\text{new}}) + x^{\text{new}} = J' - \mu(y) + x^{\text{new}}.$$

  Since μ is a feasible-exchange bijection, $J' - \mu(y) + y \in \mathcal{I}$. By downward-closedness, $J' - \mu(y) \in \mathcal{I}$. By definition of the free extension, $(J' - \mu(y)) + x^{\text{new}} \in \mathcal{I}'$.

Therefore, a feasible-exchange bijection exists for I' and J'.

Case 3: J' contains x^{new} while I' is a basis of $\mathcal{M}$. This case is analogous to Case 2.

Case 4: both I' and J' contain x^{new}. Similarly to Case 2, we write $I' = I + x^{\text{new}}$ and find a basis $I + y$ of $\mathcal{M}$. Let $\mu : I + y \to J'$ be a feasible-exchange bijection guaranteed by Case 3 above, between the bases $I + y$ and J' of $\mathcal{M}'$. Now, a bijection $\mu' : I + x^{\text{new}} \to J'$ can be defined exactly as in Case 2. □

References

Anari, N., Mai, T., Gharan, S. O., & Vazirani, V. V. (2018). Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. In *Proceedings of SODA-18*, pp. 2274–2290. SIAM.

Babaioff, M., Ezra, T., & Feige, U. (2021). Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of AAAI-21*, pp. 5119–5126.

Barman, S., & Verma, P. (2021). Existence and computation of maximin fair allocations under matroid-rank valuations. In *Proceedings of AAMAS-21*, pp. 169–177.

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

Bei, X., Garg, J., Hoefer, M., & Mehlhorn, K. (2017). Earning limits in fisher markets with spending-constraint utilities. In *Proceedings of SAGT-17*, pp. 67–79. Springer.

Bei, X., Igarashi, A., Lu, X., & Suksompong, W. (2022). The price of connectivity in fair division. *SIAM Journal on Discrete Mathematics*, *36*(2), 1156–1186.

Benabbou, N., Chakraborty, M., Igarashi, A., & Zick, Y. (2021). Finding fair and efficient allocations for matroid rank valuations. *ACM Transactions on Economics and Computation*, *9*(4), 1–41.

Bilò, V., Caragiannis, I., Flammini, M., Igarashi, A., Monaco, G., Peters, D., Vinci, C., & Zwicker, W. S. (2018). Almost Envy-Free Allocations with Connected Bundles. In Blum, A. (Ed.), *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, Vol. 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 14:1–14:21, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Biswas, A., & Barman, S. (2018). Fair division under cardinality constraints. In *Proceedings of IJCAI-18*, pp. 91–97. The full version, dated 19 Oct 2020, is available at https://arxiv.org/abs/1804.09521v3.

Biswas, A., & Barman, S. (2019). Matroid constrained fair allocation problem. In *Proceedings of AAAI-19*, Vol. 33, pp. 9921–9922.

Bouveret, S., Cechlárová, K., Elkind, E., Igarashi, A., & Peters, D. (2017). Fair division of a graph. In *Proceedings of IJCAI-17*, pp. 135–141.

Brualdi, R. A. (1969). Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society*, *1*(2), 161–167.

Brualdi, R. A. (1971). Induced matroids. *Proceedings of the American Mathematical Society*, *29*(2), 213–221.

Brualdi, R. A., & Scrimger, E. B. (1968). Exchange systems, matchings, and transversals. *Journal of Combinatorial Theory*, *5*(3), 244–257.

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

Budish, E., Cachon, G. P., Kessler, J. B., & Othman, A. (2017). Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, *65*(2), 314–336.

Calinescu, G., Chekuri, C., Pál, M., & Vondrák, J. (2011). Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, *40*(6), 1740–1766.

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

Chaudhury, B. R., Garg, J., & Mehlhorn, K. (2020). Efx exists for three agents. In *Proceedings of EC-20*, pp. 1–19.

Dror, A., Feldman, M., & Segal-Halevi, E. (2021). On fair division under heterogeneous matroid constraints. In *Proceedings of the AAAI-21*, pp. 5312–5320.

Edmonds, J. (1965). Minimum partition of a matroid into independent subsets. *J. Res. Nat. Bur. Standards Sect. B*, *69*, 67–72.

Edmonds, J. (1970). Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization — Eureka, You Shrink!*, pp. 11–26. Springer. reprinted 2003.

Ferraioli, D., Gourvès, L., & Monnot, J. (2014). On regular and approximately fair allocations of indivisible goods. In *Proceedings of AAMAS-14*, pp. 997–1004.

Frank, A. (1981). A weighted matroid intersection algorithm. *Journal of Algorithms*, *2*(4), 328–336.

Gafni, Y., Huang, X., Lavi, R., & Talgam-Cohen, I. (2021). Unified fair allocation of goods and chores via copies.. arXiv preprint 2109.08671.

Gan, J., Li, B., & Wu, X. (2021). Approximately envy-free budget-feasible allocation.. arXiv preprint 2106.14446.

Garg, J., Hoefer, M., & Mehlhorn, K. (2018). Approximating the nash social welfare with budget-additive valuations. In *Proceedings of SODA-18*, pp. 2326–2340. SIAM.

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

Gourvès, L., & 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., & Tlilane, L. (2013). A protocol for cutting matroids like cakes. In *Proceedings of WINE-13*, pp. 216–229. Springer.

Gourvès, L., Monnot, J., & Tlilane, L. (2014). Near fairness in matroids. In *Proceedings of ECAI-14*, Vol. 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 393–398. IOS Press.

Hummel, H., & Hetland, M. L. (2022a). Fair allocation of conflicting items. *Autonomous Agents and Multi-Agent Systems*, *36*(1), 1–33.

Hummel, H., & Hetland, M. L. (2022b). Maximin shares under cardinality constraints. In *Proceedings of EUMAS-22*, pp. 188–206. Springer.

Jojic, D., Panina, G., & Zivaljevic, R. (2021). Splitting necklaces, with constraints. *SIAM Journal on Discrete Mathematics*, *35*(2), 1268–1286.

Klaus, B., Manlove, D. F., & Rossi, F. (2016). *Matching under preferences*. Cambridge University Press.

Kyropoulou, M., Suksompong, W., & Voudouris, A. A. (2020). Almost envy-freeness in group resource allocation. *Theoretical Computer Science*, *841*, 110–123.

Lehmann, B., Lehmann, D., & Nisan, N. (2001). Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pp. 18–28.

Li, B., Li, M., & Zhang, R. (2021). Fair scheduling for time-dependent resources. *Advances in Neural Information Processing Systems*, *34*.

Li, Z., & Vetta, A. (2021). The fair division of hereditary set systems. *ACM Transactions on Economics and Computation (TEAC)*, *9*(2), 1–19.

Lian, J. W., Mattei, N., Noble, R., & Walsh, T. (2018). The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *Proceedings of AAAI-18*, pp. 1138–1145. AAAI Press.

Lipton, R. J., Markakis, E., Mossel, E., & Saberi, A. (2004). On approximately fair allocations of indivisible goods. In *Proceedings of EC-04*, pp. 125–131.

Long, C., Wong, R. C.-W., Peng, Y., & Ye, L. (2013). On good and fair paper-reviewer assignment. In *Proceegins of the IEEE International Conference on Data Mining*, pp. 1145–1150. IEEE.

Mackin, E., & Xia, L. (2016). Allocating indivisible items in categorized domains. In *Proceedings of IJCAI-16*, pp. 359–365.

Nyman, K., Su, F. E., & 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.

Oxley, J. G. (2006). *Matroid theory*, Vol. 3. Oxford University Press, USA.

Rado, R. (1942). A theorem on independence relations. *The Quarterly Journal of Mathematics*, *1*(1), 83–89.

Robinson, G., & Welsh, D. (1980). The computational complexity of matroid properties. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 87, pp. 29–45. Cambridge University Press.

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

Shoshan, H., Segal-Halevi, E., & Hazon, N. (2023). Efficient nearly-fair division with capacity constraints. In *Proceedings of AAMAS-23 (to appear)*. arXiv preprint 2205.07779.

Sikdar, S., Adalı, S., & Xia, L. (2017). Mechanism design for multi-type housing markets. In *Proceedings of AAAI-17*, pp. 684–690.

Sikdar, S., Adalı, S., & Xia, L. (2019). Mechanism design for multi-type housing markets with acceptable bundles. In *Proceedings of AAAI-19*, Vol. 33, pp. 2165–2172.

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

Suksompong, W. (2021). Constraints in fair division. *ACM SIGecom Exchanges*, *19*(2), 46–61.

Turner, J. (2015a). Faster maximium priority matchings in bipartite graphs. Tech. rep.. arXiv preprint 1512.09349.

Turner, J. (2015b). Maximum priority matchings. Tech. rep.. arXiv preprint 1512.08555.

Vondrák, J. (2008). Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of STOC-08*, pp. 67–74.

Wu, X., Li, B., & Gan, J. (2021). Budget-feasible maximum nash social welfare is almost envy-free.. In *Proceedings of IJCAI-21*, pp. 1–16.