Constrained Fair Allocations via Reductions to Simpler Matroids

Benjamin Cookson Nisarg Shah

University of Toronto
{bcookson, nisarg}@cs.toronto.edu

Abstract

We study fair allocation of indivisible goods under additive valuations and matroid constraints. A challenging open question is whether an envy-free up to one good (EF1) allocation exists under every matroid that admits a feasible allocation. The state-of-the-art result by Biswas and Barman [2018] positively resolves this question for partition matroids (cardinality constraints).

In this work, we make progress on this open question by leveraging a technique of *matroid reductions*. At a high level, all our results involve taking a complex matroid, and breaking it down into a much simpler matroid in such a way that any allocation that is feasible subject to the simpler matroid constraint is also feasible subject to the original matroid constraint.

Using this technique, we are able to solve the open question for several new classes of matroids. Our first result positively resolves it for laminar matroids, which generalize partition matroids, when there are three agents. Our technique involves reducing EF1 existence under all laminar matroids to EF1 existence (with a mild additional condition) under a single finite-sized key laminar matroid, and establishing the latter by case analysis. We show that our technique somewhat extends to four agents, reducing the problem to EF1 existence under two finite-sized laminar matroids, although we are unable to establish that. We also leverage recent matroid decomposition results to establish EF1 existence under broader classes of matroids. We are able to show that EF1 allocations always exist under transversal matroids that admit a feasible allocation, and show existence for graphic matroids and gammoids assuming conditions stronger than the existence of a feasible allocation.

1 Introduction

Fair allocation of indivisible goods has long been a central problem in economics and computer science. In its most basic form, it asks how to partition a set of m indivisible goods among n agents with additive valuations, i.e., when their value for a set of goods is their total value for the individual goods in that set.

While this basic model offers significant algorithmic insights on fairness, it suffers from limited direct applicability as many real-world settings impose additional structure on the desired allocation. For example, a museum may wish to allocate exhibit items between its several branches, but must additionally ensure that no branch receives too many items from any single category (e.g., paintings or sculptures). Mathematically, this is depicted as a *partition matroid constraint*, where the bundle allocated to each branch (agent) must be an independent set of a partition matroid over the exhibit items (goods) induced by the categories and their upper limits. Some constraints are more complex. When a hospital assigns shifts to nurses, it may wish to not only limit the number of shifts on any given day of the week, but also the total number of shifts allocated in the entire week. Similarly, when a conference matches reviewers to submissions, it may model having multiple reviewers per submission by creating copies of the submission but enforcing that each reviewer gets at most one copy of each submission, in addition to limiting the total review load. These examples can be viewed as *laminar matroid constraints*, which generalize partition matroids by imposing upper bounds with respect to a laminar family of sets in which every pair of sets are either disjoint or one is contained in the other. Assigning courses to students at a university can involve additional layers as students face total course load limits, departmental caps, and sometimes fine-grained constraints on courses from different sub-fields, yielding even richer laminar structures.

A substantial literature studies fair allocation under such constraints. A prominent open question is a strikingly simple one: *Does an envy-free up to one good (EF1) allocation always exist subject to matroid constraints (that still admit a feasible allocation)?* EF1 is arguably the most compelling fairness criterion for allocating indivisible goods [Lipton *et al.*, 2004; Budish, 2011; Caragiannis *et al.*, 2019], which demands that no agent strictly prefer the bundle assigned to another agent with her favorite good removed to her own bundle. This criterion is surprisingly flexible: it can be attained in conjunction with Pareto optimality [Caragiannis *et al.*, 2019] and $2/3$-maximin share fairness [Akrami and Rathi, 2025], allows achieving exact envy-freeness in expectation by randomizing over EF1 allocations [Aziz *et al.*, 2023], and continues to exist under monotone valuations [Lipton *et al.*, 2004] or when allocating (undesirable) chores together with goods [Aziz *et al.*, 2022]. Therefore, it is striking that its existence remains a major unresolved question in the literature on constrained fair division.

Progress on this question has so far been limited, with the state-of-the-art result by Biswas and Barman [2018] proving EF1 existence under partition matroid constraints and additive valuations. See Section 1.2 for an overview of further related work. In this paper, we make progress on this fundamental problem by proving several new existence results for laminar (and broader classes of) matroid constraints using a technique called *partition matroid reductions*.

1.1 Our Contributions

In this work, we show that EF1 allocations exist for several classes of matroid constraints that were previously unknown. These results all revolve around a common technique, in which we take a complex matroid, and reduce it to a simple matroid where existence of EF1 is already known (such as a partition matroid) or where existence of EF1 is quite simple to prove.

Our first result, presented in Section 3, positively resolves the EF1 existence question for three agents under laminar matroid constraints. Specifically, we show that whenever a laminar matroid admits a feasible allocation to three agents, it also admits a feasible EF1 allocation to three agents. We demonstrate that the problem of EF1 existence under any laminar matroid reduces to establishing existence of a slightly stronger condition than EF1 under a specific, finite-sized *key laminar matroid*. We are able to prove such a condition for this key matroid, and we show how this result combines with the algorithm of Biswas and Barman [2018] to yield a polynomial-time solution for this problem.

Additionally, in Appendix B, we investigate how our techniques extend to the problem of laminar matroid constrained instances with more than 3 agents. We show that our decomposition technique extends to the four-agent setting, effectively reducing the general problem to checking two specific finite-sized laminar matroids. However, resolving these base cases remains an open challenge. In fact, we show that the current techniques we used for solving the key matroids in the three-agent case are insufficient for $n = 4$.

Finally, we examine fair allocation under several other classes of matroid constraint. We leverage recent matroid theory results from Bérczi *et al.* [2021], which imply that certain classes of matroids can always be reduced to partition matroids in such a way that finding an EF1 allocation for that partition matroid will result in an EF1 allocation for the original matroid. This reduction, along with the algorithm of Biswas and Barman [2018] for allocating partition matroids, provides a powerful combination that allows us to turn these complex matroids into partition matroids and then solve for EF1 allocations. However, these reductions come at some cost, if the original matroid admits a feasible allocation to n agents, then the resulting partition matroid may only admit a feasible allocation for some $f(n) \geqslant n$ agents. Specifically, the results of Bérczi *et al.* [2021] give the following bounds for different classes of matroid:

- If the matroid is a *transversal matroid*, then $f(n) = n$.
- If the matroid is a *graphic matroid*, then $f(n) = 2n - 1$.
- If the matroid is a *gammoid*, then $f(n) = 2n - 2$.

For transversal matroids, this result fully resolves the EF1 matroid conjecture for this class, showing that if a given transversal matroid admits any feasible allocation to n agents, then an EF1 allocation to n agents also exists.

For graphic matroids and gammoids, this result resolves a slightly weaker version of the EF1 matroid conjecture, showing that if such a matroid admits a feasible allocation to n agents, then a feasible EF1 allocation to $f(n) > n$ agents always exists.

1.2 Related Work

As stated previously, Biswas and Barman [2018] initiated the search for EF1 under matroid constraints. In their original paper, they presented an elegant, polynomial-time algorithm that finds an EF1 allocation subject to partition matroid constraints. In the same paper, Biswas and Barman showed that when agents have identical valuations, an EF1 allocation always exists under laminar matroid constraints. Later, in [Biswas and Barman, 2019], the same authors extended the latter result to work under all base-orderable matroid constraints, a much broader class than laminar.

Going beyond base-orderable matroids when agents have identical valuations, Akrami *et al.* [2025] resolve the matroid equitability conjecture, and as a consequence, show that an EF1 allocation exists when agents have identical tri-valued valuations. Further, they show that if Gabow's Conjecture [Gabow, 1976] (a long-standing conjecture in matroid theory) is true, then an EF1 allocation exists under identical valuations subject to any matroid constraint.

When agents have non-identical valuation functions, Cookson *et al.* [2025] show that subject to any matroid constraint, the feasible allocation that maximizes Nash Welfare achieves a $1/2$ approximation of EF1 while also achieving the efficiency notion of Pareto Optimality. Another paper that studies matroid constrained allocations that are simultaneously fair and efficient is [Shoshan *et al.*, 2022], who show that an EF1 and PO allocation always exists for two agents under partition matroid constraints.

Dror *et al.* [2023] also study EF1 under matroid constraints. Their main results are regarding heterogeneous matroid constraints, where each agent's bundle is constrained by a different matroid. When there is a single matroid constraint for all agents, they show that there always exists a complete EF1 allocation to three agents subject to any base-orderable matroid constraint when agents have (non-identical) binary valuations. Particularly relevant to our work, Dror *et al.* [2023] also point out that for any class of matroid where it is known that an EF1 allocation always exists subject to identical preference functions, the existence of an EF1 allocation for two agents with different preferences follows from this and a simple cut-and-choose algorithm. This establishes that EF1 always exists for two agents subject to base-orderable matroids (and thus also laminar matroids).

Finally, many other classes of constraints other than matroid constraints have been studied in the context of fair division. For a comprehensive overview of this literature, we direct the reader to [Suksompong, 2021].

Independent Work: Independently of this work, Equbal [2026] also proved the existence of an EF1 allocation to 3 agents under a laminar matroid constraint. Our techniques for proving this result differ slightly, but both hinge on identifying the same structural bottlenecks that make laminar matroids hard to solve. Both our algorithms highlight that the problem can essentially be reduced to finding an allocation with certain properties for a very specific laminar constrained instance with 6 goods (named the *key matroid* in this paper, and discussed in detail in Section 3.4). Both this paper and the paper of Equbal [2026] provide essentially the same proof showing that the required allocation over these key matroid instances always exists. Then, with a subroutine for solving key matroid instances in hand, both our algorithm and that of Equbal [2026] rely on modifications of the partition matroid allocation algorithm of Biswas and Barman [2018] that uses this subroutine in order to produce the final fair allocation.

Where the two results differ is in their details and presentation. The algorithm of Equbal [2026] takes a bottom-up approach, starting at the leaves of the laminar family and solving these leaves in a similar fashion to how Biswas and Barman [2018] solve individual partitions in their algorithm. Then, observing that after solving these partitions, any leftover unallocated items will need to be allocated subject to key matroid constraints in order to maintain feasibility, it completes its allocation using the designed subroutine.

In contrast, our algorithm takes a top-down approach. We first systematically decompose the input laminar matroid constraint into a new constraint that is a collection of disjoint uniform matroids and key matroids. We solve this reduced matroid by applying the algorithm of Biswas and Barman [2018] directly on the uniform matroids, and use the new subroutine to resolve the key matroids.

Ultimately, our goal with our laminar matroid algorithm and the subsequent results in Section 4 is to highlight the usefulness of reducing complex matroid constraints to simple matroid constraints, then finding EF1 allocations subject to the simple constraints. With this in mind, we believe our top-down approach to the laminar problem highlights this, and helps give context to exactly what makes laminar matroid constraints harder than partition matroid constraints. We hope that this perspective can be used to resolve laminar matroids for more than 3 agents, and to generalize to larger classes of matroids than just the ones discussed in this paper.

2 Preliminaries

For any $r \in \mathbb{N}$, define $[r] := \{1, 2, \ldots, r\}$. Let $N = [n]$ be a set of *agents*, and M be a set of m *(indivisible) goods*. Each agent i has a valuation function $v_i : M \to \mathbb{R}_{\geqslant 0}$, where $v_i(g)$ is her value for good g. We assume additive valuations: with slight abuse of notation, the value of agent i for a set of goods $S \subseteq M$ is $v_i(S) := \sum_{g \in S} v_i(g)$. Let $v = (v_1, \ldots, v_n)$ be the *valuation profile*. In an *allocation* $A = (A_1, \ldots, A_n)$, $A_i \subseteq M$ is the bundle of goods assigned to agent i and $A_i \cap A_j = \emptyset$ for all distinct $i, j \in N$. We say that A is *complete* if $\cup_{i \in N} A_i = M$. The utility to agent i under this allocation is $v_i(A_i)$.

2.1 Matroid Constraints

We are interested in *matroid-constrained* fair division, in which we are allowed to assign a bundle to an agent only if the bundle is an independent set of a provided matroid over the set of goods.

Formally, a matroid over the set of goods M is defined as $(M, \mathcal{I})$, where $\mathcal{I} \subseteq 2^M$ satisfies: (a) $\emptyset \in \mathcal{I}$; (b) *(hereditary property)* if $S \in \mathcal{I}$ and $S' \subseteq S$, then $S' \in \mathcal{I}$; and (c) *(exchange property)* if $S, T \in \mathcal{I}$ and $|T| > |S|$, then there exists $g \in T \setminus S$ such that $S \cup \{g\} \in \mathcal{I}$.

A matroid $(M, \mathcal{I})$ is called k-colorable if there exists a partition $(S_1, \ldots, S_k)$ of M such that $S_i \in \mathcal{I}$ for all $i \in [k]$. An allocation A is called *feasible* if $A_i \in \mathcal{I}$ for all agents $i \in N$. Hence, in a matroid-constrained fair division instance, a feasible allocation to n agents exists if and only if the matroid is n-colorable. Hereinafter, when we refer to an allocation, we mean a feasible allocation unless explicitly stated otherwise.

We will refer to the following popular families of matroids in this paper.

- **Uniform matroid:** Given the ground set M and an integer capacity $c \in \mathbb{N}$, set $\mathcal{I} = \{S \subseteq M : |S| \leqslant c\}$. This is one of the simplest matroids.
- **Partition matroid (cardinality constraints):** Let $\mathcal{M} = (M_1, \ldots, M_k)$ be a partition of M into disjoint sets ("categories") and let $u : \mathcal{M} \to \mathbb{N}$ provide their corresponding capacities. Then, set $\mathcal{I} = \{S \subseteq M : |S \cap M_j| \leqslant u(M_j) \text{ for all } j \in [k]\}$. Hence, a partition matroid is the intersection of several uniform matroids.
- **Laminar matroid:** Let $\mathcal{L}$ be a *laminar family* of subsets of M (i.e., for all $X, Y \in \mathcal{L}$, $X \subseteq Y$, $Y \subseteq X$, or $X \cap Y = \emptyset$) and $u : \mathcal{L} \to \mathbb{N}$ provide their corresponding capacities. Then, set $\mathcal{I} = \{S \subseteq M : |S \cap X| \leqslant u(X) \text{ for all } X \in \mathcal{L}\}$.
- **Transversal matroid:** Let G be an undirected bipartite graph with parts M and U. Then, set $\mathcal{I}$ to be the set of all subsets of M that can be covered by a matching of G. Formally, given a set U and its subsets $B_1, \ldots, B_m$, let $\mathcal{I}$ be the collection of sets $S \subseteq M$ for which there exists an injective function $f : S \to U$ with $f(i) \in B_i$ for all $i \in S$.
- **Gammoid:** Let $D = (V, E)$ be a directed graph and $S, T \subseteq V$ be disjoint sets of source and terminal vertices, respectively. Then, let $\mathcal{I}$ be the set of all sets $I \subseteq T$ such that there exist $|I|$ vertex-disjoint paths connecting S to I.
- **Graphic Matroid:** Let $G = (V, M)$ be an undirected graph (note that the ground set M is now the set of edges). Then, let $\mathcal{I}$ be the set of all sets $S \subseteq M$ that do not contain a cycle (i.e., are forests).

These classes are related as follows:

- Uniform $\subset$ Partition $\subset$ Laminar $\subset$ Gammoid;
- Uniform $\subset$ Transversal $\subset$ Gammoid.

As our main result concerns laminar matroids, let us introduce additional notation for them. Consider the laminar matroid induced by the ground set M, a laminar set family $\mathcal{L}$ consisting of subsets of M, and a function $u : \mathcal{L} \to \mathbb{N}$ yielding capacities. We denote it as $(M, \mathcal{L}, u)$ and write its family of independent sets as $\mathcal{I}(M, \mathcal{L}, u)$, dropping the arguments in brackets when the matroid is clear from the context. For each $S \in \mathcal{L}$, define two quantities.

- Let $D(S) = \{S' \in \mathcal{L} : S' \subsetneq S\}$. We will refer to these as the *descendants* of S.
- Let $C(S)$ be the set of inclusion-wise maximal elements of $D(S)$, i.e., containing those $S' \in D(S)$ for which there is no $S'' \in D(S)$ with $S' \subsetneq S''$. We will refer to these as the *children* of S.

The reasoning behind this nomenclature will become clear in Section 3.1.

Finally, for all these constrained fair allocation problems, we define *fairness* as Envy-freeness up to one good (EF1).

Definition 1 (Envy-freeness up to one good (EF1))**.** An allocation A is *envy-free up to one good* if, for every pair of agents $i, j \in N$, either $A_j = \emptyset$ or $v_i(A_i) \geqslant v_i(A_j \setminus \{g\})$ for some $g \in A_j$. In words, agent i should not envy agent j, after excluding some good from agent j's bundle.

3 Laminar Matroids With 3 Agents

In this section, we prove our main result.

Theorem 1. *For $n = 3$ agents with additive valuations and laminar matroid constraints, a complete and feasible EF1 allocation exists whenever a complete and feasible allocation exists (i.e., whenever the matroid is 3-colorable), and such an allocation can be computed in polynomial time.*

First, we introduce a technique to reduce the given matroid to a simpler matroid subject to which a fair allocation can be found.

Definition 2 (Feasibility-Preserving Operation)**.** We say that an operation which takes an n-colorable laminar matroid $(M, \mathcal{L}, u)$ as input and outputs $(M, \mathcal{L}', u')$ is *feasibility-preserving* if:

1. (Validity) $(M, \mathcal{L}', u')$ is a laminar matroid, i.e., $\mathcal{L}'$ is a laminar set family;
2. (Rank Preservation) $(M, \mathcal{L}', u')$ is n-colorable, i.e., the new matroid also admits a complete and feasible allocation; and
3. (Independent Set Shrinkage) $\mathcal{I}(M, \mathcal{L}', u') \subseteq \mathcal{I}(M, \mathcal{L}, u)$, i.e., every independent set of the new matroid is also an independent set of the original matroid.

The Rank Preservation condition ensures that we do not rule out the existence of a feasible and fair allocation to n agents by ruling out the existence of a feasible allocation to n agents in the first place. So long as a feasible allocation exists, our result will ensure that a feasible and fair allocation would exist as well. The Independent Set Shrinkage condition ensures that any feasible allocation subject to the resulting matroid remains feasible subject to the original matroid. In our proof, we perform a sequence of feasibility-preserving operations on the given laminar matroid, and then find an EF1 allocation subject to the resulting matroid, which is then guaranteed to be an EF1 allocation subject to the original matroid as well.

In more detail, our proof for Theorem 1 consists of four stages.

In **Stage 1**, we introduce a feasibility-preserving operation to completely *pack* the laminar matroid—specifically, this ensures that for each $S \in \mathcal{L}$ who has descendants, each good in S appears in some descendant of S.

In **Stage 2**, we introduce a set of feasibility-preserving operations, which are applied repeatedly to *simplify* the packed matroid from Stage 1, unless no such operation can be applied.

In **Stage 3**, we show that the packed and simplified matroid resulting from Stage 2 consists of connected components that are either uniform matroids or a special laminar matroid over 6 goods, which we term *key matroid*. Using a result of Biswas and Barman [2018], the existence of a feasible EF1 allocation subject to all laminar matroids (over any number of goods) reduces to the existence of a feasible EF1 allocation (with an additional ordered no-envy condition) subject to just this single key matroid.

In **Stage 4**, we "solve" the key matroid, proving that it always admits the desired allocation regardless of the agent valuations.

Our method is constructive and results in a polynomial-time algorithm to find an EF1 allocation. Let us now understand all four stages in detail.

3.1 Stage 1: Packing

Definition 3 (Packed Laminar Set Family)**.** We say that a laminar set family $\mathcal{L}$ over a ground set M is **packed** if, for every $S \in \mathcal{L}$, either $D(S) = \emptyset$ or $\bigcup D(S) = S$.

The following is a more useful characterization of the packed condition.

Lemma 1. *A laminar set family $\mathcal{L}$ is packed if and only if, for each $S \in \mathcal{L}$, either $C(S) = \emptyset$ or $C(S)$ forms a disjoint partition of S.*

Proof. The "if" direction is trivial. If $C(S) = \emptyset$, then $D(S) = \emptyset$, and if $C(S)$ forms a disjoint partition of S, then, $\bigcup D(S) \supseteq \bigcup C(S) = S$, which implies $\bigcup D(S) = S$.

For the "only if" direction, take any $S \in \mathcal{L}$. If $C(S) = \emptyset$, we are done. Suppose $C(S) \neq \emptyset$. First, we observe that $\bigcup C(S) = S$. Pick any $g \in S$. Since $\bigcup D(S) = S$, there exists $S' \in D(S)$ such that $g \in S'$. Pick any inclusion-wise maximal (e.g., maximum-cardinality) set in $D(S)$ that contains g. Then, $S' \in C(S)$. Hence, $g \in \bigcup C(S)$. Since this holds for any $g \in S$, we have $\bigcup C(S) = S$. Next, we observe that for any $S', T' \in C(S)$, $S' \cap T' = \emptyset$. The laminar structure implies that $S' \cap T' = \emptyset$, $S' \subset T'$, or $T' \subset S'$. Then, maximality of every child in $C(S)$ implies that it must be $S' \cap T' = \emptyset$. □

The structure in Lemma 1 allows one to view a packed laminar set family $\mathcal{L}$ as a forest, with each connected component being a rooted tree. Specifically, the sets in $\mathcal{L}$ are the vertices, and for each $S \in \mathcal{L}$, $C(S)$ are the *children* of S and $D(S)$ are the **descendants** of S (i.e., leaves are precisely those $S \in \mathcal{L}$ with $D(S) = \emptyset$).

Finally, we can introduce packed laminar matroids.

Definition 4 (Packed Laminar Matroid)**.** We say that a laminar matroid $(M, \mathcal{L}, u)$ is *n-packed* if:

- $\mathcal{L}$ is packed, and
- $u(S) = \lceil |S|/n \rceil$ for all $S \in \mathcal{L}$.

The intuition behind the value of $u(S)$ is as follows. It is known that n-colorability demands $u(S) \geqslant \lceil |S|/n \rceil$ for all $S \in \mathcal{L}$. So, if we impose the more stringent equality condition, a feasible allocation subject to that would remain feasible subject to the original matroid.

Let us now introduce the feasibility-preserving operation which can pack any laminar matroid.

Definition 5 (Laminar Packing)**.** Given a positive integer n and an n-colorable laminar matroid $(M, \mathcal{L}, u)$, construct a laminar matroid $(M, \mathcal{L}', u')$ as follows.

- Let $\mathcal{L}' = \mathcal{L}$.
- For each $S \in \mathcal{L}$, if $0 < \left|\bigcup D(S)\right| < |S|$, then add $S^* = S \setminus \bigcup D(S)$ to $\mathcal{L}'$.
- Set $u'(S) = \lceil |S|/n \rceil$ for all $S \in \mathcal{L}'$.

From the definition, it is clear that applying laminar packing on any laminar matroid yields a packed laminar matroid. The following result shows that it does so while preserving feasibility.

Lemma 2. *The laminar packing operation is feasibility-preserving.*

Proof. Suppose we apply laminar packing to an n-colorable matroid $(M, \mathcal{L}, u)$ to produce a matroid $(M, \mathcal{L}', u')$.

First, we prove that $\mathcal{L}'$ is a laminar set family. Adding each S^* preserves the laminar structure: by definition, any good $g \in S^*$ is not in any descendant of S, so the only sets that could contain g are strict supersets of S. Thus, S^* is a strict subset of S, hence also of its supersets, and has no overlap with any other set.

Next, we prove the rank preservation condition, i.e., that $(M, \mathcal{L}', u')$ is also n-colorable. It is well known that a laminar matroid $(M, \mathcal{L}, u)$ is n-colorable if and only if $u(S) \geqslant \lceil |S|/n \rceil$ for all $S \in \mathcal{L}$ (for completeness, we provide a short proof of this fact in Lemma 10). Since this condition still holds for $(M, \mathcal{L}', u')$, it is also n-colorable.

Finally, we prove the independent set shrinkage condition, i.e., $\mathcal{I}(M, \mathcal{L}', u') \subseteq \mathcal{I}(M, \mathcal{L}, u)$. For contradiction, suppose there exists $I \in \mathcal{I}(M, \mathcal{L}', u') \setminus \mathcal{I}(M, \mathcal{L}, u)$. Since $I \notin \mathcal{I}(M, \mathcal{L}, u)$, there must exist $S \in \mathcal{L}$ for which $|I \cap S| > u(S)$. However, since the laminar packing operation only adds sets, $S \in \mathcal{L} \subseteq \mathcal{L}'$. Then, $|I \cap S| > u(S) \geqslant \lceil |S|/n \rceil = u'(S)$, contradicting the fact that $I \in \mathcal{I}(M, \mathcal{L}', u')$. □

3.2 Stage 2: Simplification

Once the input laminar matroid is packed in Stage 1, we apply several operations on the resulting matroid, which are not only feasibility-preserving, but also keep the resulting matroid packed.

Definition 6 (Packing-Preserving Operation)**.** We say that an operation that takes a packed laminar matroid $(M, \mathcal{L}, u)$ as input and returns a laminar matroid $(M, \mathcal{L}', u')$ as output is *packing-preserving* if $(M, \mathcal{L}', u')$ is also packed.

We will introduce four such operations: *splitting*, *child pruning*, *parent pruning*, and *partition extraction*. All operations will take an n-colorable laminar matroid $(M, \mathcal{L}, u)$ as input and return a matroid $(M, \mathcal{L}', u')$. To prove that they are feasibility-preserving and packing-preserving, we will need to show that they return a laminar matroid that is packed, n-colorable, and satisfies $\mathcal{I}(M, \mathcal{L}', u') \subseteq \mathcal{I}(M, \mathcal{L}, u)$. Our algorithm will successively keep applying these operations until none of them can be applied anymore.

Definition 7 (Splitting)**.** If there exists an $S \in \mathcal{L}$ and $S_1 \subsetneq S$ such that (i) $|S_1|$ is a positive multiple of n, and (ii) for each $C \in C(S)$, either $C \subseteq S_1$ or $C \cap S_1 = \emptyset$, then replace S with two disjoint sets S_1 and $S_2 = S \setminus S_1$ (i.e., $\mathcal{L}' = \mathcal{L} \cup \{S_1, S_2\} \setminus \{S\}$) and set the new capacities as $u'(S_1) = |S_1|/n$ and $u'(S_2) = \lceil |S_2|/n \rceil$ (with $u'(T) = u(T)$ for all $T \in \mathcal{L} \setminus \{S\}$).

In the tree form, this amounts to replacing node S with nodes S_1 and S_2, making them children of the parent of S, and placing each child C of S (and the subtree below it) under S_1 if $C \subseteq S_1$ and under S_2 otherwise.

Lemma 3. *The splitting operation is feasibility-preserving and packing-preserving.*

Proof. First, we show that the resulting matroid remains laminar. Since the laminar structure holds for all the pairs of sets that were part of $\mathcal{L}$, we simply need to establish it for pairs of sets in which one or both sets come from $\{S_1, S_2\}$. It holds for the (S_1, S_2) pair as $S_1 \cap S_2 = \emptyset$. Take the pair (S_1, T) for some $T \in \mathcal{L} \setminus \{S\}$. If $T \in D(S)$, then by definition we have either $T \subseteq S_1$ or $T \subseteq S_2$ (in which case $T \cap S_1 = \emptyset$). If $S \subseteq T$, then $S_1 \subseteq T$. Finally, if $S \cap T = \emptyset$, then $S_1 \cap T = \emptyset$. Hence, the laminar structure holds for all pairs (S_1, T) where $T \in \mathcal{L} \setminus \{S\}$. A symmetric argument holds when replacing S_1 with S_2. Hence, $\mathcal{L}'$ remains laminar.

Next, $\mathcal{L}'$ remains packed because $C(S)$ is a disjoint partition of S, so, by construction, we must have that $C(S_1)$ and $C(S_2)$ are disjoint partitions of S_1 and S_2 as well, implying $\bigcup D(S_1) = S_1$ and $\bigcup D(S_2) = S_2$. For any $T \in \mathcal{L}$ such that $S \subsetneq T$, we now have $S_1, S_2 \in D(T)$, so $\bigcup D(T) = T$ holds as previously. Finally, $\bigcup D(T)$ remains unaffected for any other $T \in \mathcal{L}$.

Finally, we show independent set shrinkage: $\mathcal{I}(M, \mathcal{L}', u') \subseteq \mathcal{I}(M, \mathcal{L}, u)$. Suppose for contradiction that there exists $I \in \mathcal{I}(M, \mathcal{L}', u') \setminus \mathcal{I}(M, \mathcal{L}, u)$. Then, there must be some $T \in \mathcal{L}$ such that $|I \cap T| > u(T)$. Since the only set in $\mathcal{L} \setminus \mathcal{L}'$ is S, it must be that case that $T = S$. However, since $I \in \mathcal{I}(M, \mathcal{L}', u')$, we have that:

$$|I \cap S_1| \leqslant u'(S_1) = |S_1|/n,$$
$$|I \cap S_2| \leqslant u'(S_2) = \lceil |S_2|/n \rceil.$$

Summing these gives:

$$\begin{aligned} |I \cap S| &= |I \cap S_1| + |I \cap S_2| \\ &\leqslant |S_1|/n + \lceil |S_2|/n \rceil \\ &= \lceil (|S_1| + |S_2|)/n \rceil = \lceil |S|/n \rceil. \end{aligned}$$

This contradicts $|I \cap S| > u(S)$. Crucially, the penultimate step holds because $|S_1|/n$ is an integer. □

Definition 8 (Child Pruning)**.** If there exists $S \in \mathcal{L}$ with $|S| \leqslant n$, remove all descendants of S (i.e., set $\mathcal{L}' = \mathcal{L} \setminus D(S)$ and keep $u'(T) = u(T)$ for all $T \in \mathcal{L}'$).

In the tree form, this amounts to deleting the part of the tree below S.

Lemma 4. *The child pruning operation is feasibility-preserving and packing-preserving.*

Proof. Since this operation only removes sets, $\mathcal{L}'$ remains laminar and $(M, \mathcal{L}', u')$ remains n-colorable. Because all the descendants of S are removed, the resulting matroid remains packed ($D(S)$ becomes empty while $D(T)$ remains unaffected for all other $T \in \mathcal{L}'$).

Thus, we just need to show independent set shrinkage: $\mathcal{I}(M, \mathcal{L}', u') \subseteq \mathcal{I}(M, \mathcal{L}, u)$. Suppose for contradiction that there exists $I \in \mathcal{I}(M, \mathcal{L}', u') \setminus \mathcal{I}(M, \mathcal{L}, u)$. Then, there must be some $T \in \mathcal{L}$ such that $|I \cap T| > u(T)$. Since $I \in \mathcal{I}(M, \mathcal{L}', u')$, we must have $T \in D(S)$, which implies $T \subsetneq S$. Given $|S| \leqslant n$, we get $0 < |T| < n$, so $u(T) = \lceil |T|/n \rceil = 1$ and $u(S) = 1$. But then $|I \cap T| > 1$ also implies $|I \cap S| > 1$, contradicting $I \in \mathcal{I}(M, \mathcal{L}', u')$. ☐

Definition 9 (Parent Pruning)**.** If there exists $S \in \mathcal{L}$ for which $\sum_{T \in C(S)} u(T) \leqslant u(S)$, then remove S from $\mathcal{L}$ (i.e., $\mathcal{L}' = \mathcal{L} \setminus \{S\}$ and $u'(T) = u(T)$ for all $T \in \mathcal{L}'$).

In the tree form, this amounts to deleting node S and placing every child of S directly under the parent of S.

Lemma 5. *The parent pruning operation is feasibility-preserving and packing-preserving.*

Proof. Once again, since this operation only removes sets, the resulting matroid remains laminar and n-colorable. It is also packed. To see this, note that the only node of concern is the parent T of S. Since $\bigcup C(S) = S$ and $\bigcup C(T) = T$ in the original packed matroid, $\bigcup C(T) = T$ still holds when we replace S with the sets in $C(S)$ in $C(T)$.

Thus, we just need to show independent set shrinkage: $\mathcal{I}(M, \mathcal{L}', u') \subseteq \mathcal{I}(M, \mathcal{L}, u)$. Suppose for contradiction that there exists $I \in \mathcal{I}(M, \mathcal{L}', u') \setminus \mathcal{I}(M, \mathcal{L}, u)$. Then, there must be some $T \in \mathcal{L}$ such that $|I \cap T| > u(T)$. Since only S was removed, we must have $T = S$, so $|I \cap S| > u(S)$. But for all $C \in C(S)$, we still have $|I \cap C| \leqslant u(C)$. Summing over the children, we get

$$|I \cap S| = \sum_{C \in C(S)} |I \cap C| \leqslant \sum_{C \in C(S)} u(C) \leqslant u(S),$$

where the first transition uses the fact that $C(S)$ is a disjoint partition of S, and the last transition comes from the definition of the operation. This is a contradiction. ☐

Definition 10 (Partition Extraction)**.** If there exists $S \in \mathcal{L}$ with $D(S) = \emptyset$ (leaf) and $|S| = n$, remove the elements of S from every strict superset of S (i.e., $\mathcal{L}' = \{S\} \cup \{T \setminus S : T \in \mathcal{L} \setminus \{S\}\}$), and decrease the capacity associated with every strict superset of S by 1 (i.e., $u'(T \setminus S) = u(T) - 1$ for all $T \in \mathcal{L}$ such that $S \subsetneq T$, and $u'(T) = u(T)$ for all other $T \in \mathcal{L}$).

In the tree form, this operation detaches a leaf S of size n and leaves it as a singleton connected component.

Lemma 6. *The partition extraction operation is feasibility-preserving and packing-preserving.*

Proof. First, we show that $\mathcal{L}'$ remains laminar. We need to show that for any $T_1', T_2' \in \mathcal{L}'$, $T_1' \cap T_2' = \emptyset$, $T_1' \subseteq T_2'$, or $T_2' \subseteq T_1'$. If one of the sets is S, then we have an empty intersection because the elements of S are removed from all other sets in $\mathcal{L}'$. If neither set is S, then $T_1' = T_1 \setminus S$ and $T_2' = T_2 \setminus S$ for $T_1, T_2 \in \mathcal{L}$. Since $\mathcal{L}$ is laminar, we have that $T_1 \cap T_2 = \emptyset$ (in which case $(T_1 \setminus S) \cap (T_2 \setminus S) = \emptyset$), $T_1 \subseteq T_2$ (in which case $T_1 \setminus S \subseteq T_2 \setminus S$), or $T_2 \subseteq T_1$ (in which case $T_2 \setminus S \subseteq T_1 \setminus S$). Hence, the laminar structure is established for $\mathcal{L}'$.

Next, we prove that $(M, \mathcal{L}', u')$ is n-packed. To see that $\mathcal{L}'$ is packed, note that $D(S) = \emptyset$ in $\mathcal{L}'$ by the definition of the operation. Consider any $T' = (T \setminus S) \in \mathcal{L}'$, where $T \in \mathcal{L}$. If $D(T) = \emptyset$ in $\mathcal{L}$, then $D(T') = \emptyset$ in $\mathcal{L}'$. Otherwise, $\bigcup_{X \in D(T)} X = T$ in $\mathcal{L}$, which implies that in $\mathcal{L}'$,

$$\begin{aligned}\cup_{X' \in D(T')} X' &= \cup_{X \in D(T)} (X \setminus S) \\ &= \left(\cup_{X \in D(T)} X\right) \setminus S = T \setminus S = T'.\end{aligned}$$

The next condition needed is that $u'(T') = \lceil |T'|/n \rceil$ for all $T' \in \mathcal{L}'$. When $T \cap S = \emptyset$, we have $T' = T \in \mathcal{L}$, so this holds due to $(M, \mathcal{L}, u)$ being n-packed and $u'(T') = u(T')$. When $T \supsetneq S$, we know that $T' = T \setminus S$, $|T'| = |T| - n$, and $u'(T') = u(T) - 1$. Since $(M, \mathcal{L}, u)$ is n-packed, we have $u(T) = \lceil |T|/n \rceil$. Hence,

$$\begin{aligned}u'(T') &= u(T) - 1 \\ &= \lceil |T|/n \rceil - 1 \\ &= \lceil |T|-n/n \rceil = \lceil |T \setminus S|/n \rceil = \lceil |T'|/n \rceil,\end{aligned}$$

which is desired.

Thus, we just need to show independent set shrinkage: $\mathcal{I}(M, \mathcal{L}', u') \subseteq \mathcal{I}(M, \mathcal{L}, u)$. Suppose for contradiction that there exists $I \in \mathcal{I}(M, \mathcal{L}', u') \setminus \mathcal{I}(M, \mathcal{L}, u)$. Then, there exists some $T \in \mathcal{L}$ such that $|I \cap T| > u(T)$, $S \subseteq T$, and $|I \cap T'| \leqslant u'(T') = u(T) - 1$ for $T' = T \setminus S$. However, this implies $|I \cap S| > 1$. But since $S \in \mathcal{L}$, $(M, \mathcal{L}, u)$ is n-packed, and $u(S) = 1$, this is impossible, completing the contradiction. $\square$

3.3 Stage 3: Decomposition

Next, we show that the laminar matroid that remains at the end of Stage 2 has a special structure: it forms a forest in which every connected component is either a uniform matroid or a special laminar matroid over 6 goods, which we term *key matroid*. Let us formalize this.

Definition 11 (Connected Components)**.** Define the set of top-level (i.e., inclusion-wise maximal) sets in $\mathcal{L}$ as $\mathrm{top}(\mathcal{L}) = \{S \in \mathcal{L} : \nexists T \in \mathcal{L}, S \subsetneq T\}$. Define the *connected component* induced by each $S \in \mathrm{top}(\mathcal{L})$ to be the matroid $(M', \mathcal{L}', u')$, where $M' = S$, $\mathcal{L}' = \{S\} \cup D(S)$, and $u'(T) = u(T)$ for all $T \in \mathcal{L}'$.

Note that when $D(S) = \emptyset$, the connected component has a singleton $\mathcal{L}' = \{S\}$, which induces a uniform matroid. We will prove that the only other possibility is the following.

Definition 12 (Key Matroid)**.** We say that laminar matroid $(M, \mathcal{L}, u)$ is a *key matroid* if:

- $|M| = 6$,
- $|\mathcal{L}| = 4$,
- $M \in \mathcal{L}$ with $u(M) = 2$, and
- the remaining three sets in $\mathcal{L}$ have size 2 each, form a disjoint partition of M, and have an upper bound of 1 each.

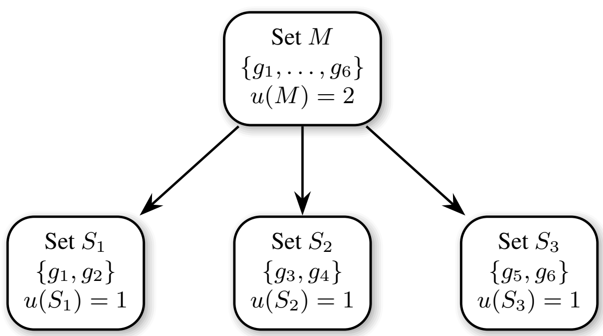

Figure 1: Structure of the key matroid

Lemma 7. *Let $(M, \mathcal{L}, u)$ be a laminar matroid that is 3-colorable and packed. If none of splitting, child pruning, parent pruning, and partition extraction operations can be applied to $(M, \mathcal{L}, u)$, then every connected component of $(M, \mathcal{L}, u)$ is either a uniform matroid or a key matroid.*

Proof. Fix a connected component $(M', \mathcal{L}', u')$ (as in the definition of connected components). Assume it is neither a uniform matroid nor a key matroid; we derive a contradiction.

Throughout, since $(M, \mathcal{L}, u)$ is 3-packed, we use $u(X) = \lceil |X|/3 \rceil$ for every $X \in \mathcal{L}$.

Step 1: Bounding leaf sizes. Let $S \in \mathcal{L}'$ be a leaf, i.e., $D(S) = \emptyset$.

- If $|S| = 3$, then *partition extraction* applies to S, contradicting the fact that no operation applies.

- If $|S| > 3$, then *splitting* applies to S: since S is a leaf, the condition involving $C(S)$ is vacuous, and we may choose any subset $S_1 \subsetneq S$ with $|S_1| = 3$ (a positive multiple of 3), resulting in a contradiction.

Hence every leaf S satisfies $|S| \leqslant 2$.

Step 2: Locating key matroids. Since the component is not a uniform matroid, it has at least one non-leaf set. Choose a non-leaf set $S \in \mathcal{L}'$ that is *lowest* so that every $C \in C(S)$ is a leaf. Given the observation in Step 1, we have $|C| \leqslant 2$ for all $C \in C(S)$.

(i) Lower bound on $|S|$. If $|S| \leqslant 3$, then *child pruning* applies to S, contradicting the fact that no operation applies. Therefore $|S| > 3$.

(ii) No singleton child. Since $\mathcal{L}$ is packed, $C(S)$ forms a disjoint partition of S (Lemma 1). If some child $C \in C(S)$ has $|C| = 1$, then regardless of whether there exists another child $C' \in C(S)$ with $|C'| = 2$ or all other children $C' \in C(S)$ have $|C'| = 1$, there would always exist a subcollection of children $\mathcal{C} \subseteq C(S)$ whose union has size $|\bigcup \mathcal{C}| = 3$ (either one 2-set plus one 1-set, or three 1-sets). Let $S_1 = \bigcup \mathcal{C}$ with $|S_1| = 3$. Since S_1 is a union of children of S, every child of S is either contained in S_1 or disjoint from it. Hence *splitting* applies to S, which is a contradiction. Thus, every child of S has size exactly 2.

Consequently, $|S|$ is even and at least 4.

Step 3: Pinning down $|S|$. Write $|S| = 2k$, where $k = |C(S)| \geqslant 2$. We rule out all values except $|S| = 6$.

- If $|S| = 4$ (so $k = 2$), then $u(S) = \lceil 4/3 \rceil = 2$ and each child C has $u(C) = \lceil 2/3 \rceil = 1$. Hence

$$\sum_{C \in C(S)} u(C) = 2 \leqslant u(S),$$

  so *parent pruning* applies to S, a contradiction.

- If $|S| \geqslant 8$ (so $k \geqslant 4$), pick any three children $C_1, C_2, C_3 \in C(S)$ and let $S_1 = C_1 \cup C_2 \cup C_3$. Then $|S_1| = 6$ is a positive multiple of 3, and again S_1 is a union of children, so *splitting* applies to S, which is a contradiction.

Therefore, $|S| = 6$ and $C(S)$ consists of exactly three children of size 2 each.

Step 4: The component must be exactly the key matroid. If S is a top-level set of the component (i.e., $S \in \text{top}(\mathcal{L})$), then the component consists of S together with its three children. With $u(S) = \lceil 6/3 \rceil = 2$ and each child having capacity $\lceil 2/3 \rceil = 1$, this is precisely the key matroid, contradicting our assumption that the component is not a key matroid.

Otherwise, S has a parent T in the component, i.e., $S \in C(T)$. Let $S_1 := S$. Then $|S_1| = 6$ is a positive multiple of 3, and, since S_1 is itself a child of T, every child of T is either contained in S_1 (namely S) or disjoint from it (all other children). Hence *splitting* applies to T, which is a contradiction.

Thus, we reach a contradiction in all cases, meaning that every connected component must be either a uniform matroid or a key matroid. □

3.4 Stage 4: Solving the Key Matroid

Finally, we "solve" the key matroid. As we will see in Section 3.5, our final algorithm requires an EF1 allocation with an additional *ordered no-envy* condition subject to the key matroid: for a given agent ordering, we need to be able to prevent envy altogether from each agent towards any agent later in the ordering.

Lemma 8. *Given $n = 3$ agents with additive valuations over $m = 6$ goods, a key laminar matroid, and an ordering over the agents, there always exists a complete and feasible EF1 allocation in which no agent envies any agent later in the ordering.*

Proof. Without loss of generality, fix the agent ordering for the ordered no-envy condition such that we want agent i to not envy agent j for any $i < j$.

We construct such an allocation by partitioning the goods into 3 bundles, then letting agent 1 pick her favorite bundle, then agent 2 her favorite bundle among the ones left, giving the remaining bundle to agent 3. This picking order guarantees the ordered no-envy condition. To ensure EF1, we design the partition to satisfy the following conditions:

1. The key matroid constraints: each bundle has size 2, and (without loss of generality) the two goods in any of the sets $\{g_1, g_2\}, \{g_3, g_4\}, \{g_5, g_6\}$ lie in different bundles.
2. The two most favorite goods of agent 2, say $\{g_a, g_b\}$, lie in different bundles.
3. The three most favorite goods of agent 3, say $\{g_x, g_y, g_z\}$, lie in different bundles.

First, let us see why this ensures EF1. Since each agent receives exactly two goods, EF1 reduces to ensuring that no agent i strictly prefers her less-preferred good from A_j to the bundle of two goods A_i that she receives. Agent 1 already does not envy anyone. For agent 2, her value for any bundle after removing her more valuable good in that bundle is at most $\min\{v_2(g_a), v_2(g_b)\}$. However, since $\{g_a, g_b\}$ are in different bundles and she goes second in the picking order, her value for her own bundle is at least $\min\{v_2(g_a), v_2(g_b)\}$, ensuring EF1. The same reasoning applies to agent 3: her value for her own bundle is at least $\min\{v_3(g_x), v_3(g_y), v_3(g_z)\}$ while her value for any bundle after removal of her favorite good from that bundle is at most this value.

It remains to show that a partition of the 6 goods satisfying the three conditions above always exists. Note that a partition into three bundles of size 2 each can be depicted as a perfect matching in an undirected graph over the goods. The third condition— the goods in $T = \{g_x, g_y, g_z\}$ lie in different bundles—reduces to seeking this perfect matching in the bipartite graph between T and $M \setminus T$. Finally, the first two conditions—the two goods in any of the sets $\{g_1, g_2\}$, $\{g_3, g_4\}$, $\{g_5, g_6\}$, and $\{g_a, g_b\}$ lie in different bundles—simply forbid the four edges between the four pairs of goods.

Starting with the complete bipartite graph $K_{3,3}$, first remove the three edges $\{g_1, g_2\}$, $\{g_3, g_4\}$, and $\{g_5, g_6\}$. In the worst case, all edges go across the two parts and get removed, leaving a graph with each vertex having degree 2, which is a 6-cycle (if not all three of those edges go across the parts, the remaining graph is a superset of a 6-cycle). This can be decomposed into two perfect matchings, so even after removing the edge between $\{g_a, g_b\}$, the perfect matching not containing this edge remains and can be selected. □

3.5 Final Algorithm

We are now ready to present our full (polynomial-time) algorithm for computing an EF1 allocation for three agents subject to a laminar matroid. The algorithm first packs the given laminar matroid, then applies the operations from Section 3.2 until none of those operations can be applied anymore. Lemma 7 shows that each connected component of the resulting matroid is either a uniform matroid or a key matroid. Our algorithm allocates the goods one connected component at a time.

In order to maintain EF1, it uses the same trick that Biswas and Barman [2018] use for computing an EF1 allocation subject to a partition matroid, where the connected components are simply uniform matroids. They maintain the property that, at any given time, in addition to the partial allocation being EF1, there is a "no-envy order" among the agents such that agents earlier in the order do not envy agents later in the order. Then, the next connected component is allocated using the round robin method [Caragiannis *et al.*, 2019] with the picking order being the reverse of the no-envy order. This ensures that envy does not compound and the resulting allocation remains EF1. The no-envy order property is then restored by running the envy-cycle elimination procedure of Lipton *et al.* [2004]. The only difference in our algorithm is to use Lemma 8 instead of the round-robin method when the connected component is a key matroid. The heavy lifting of our technique is already done by Lemma 7, which establishes the structure of the connected components, and Lemma 8, which solves the key matroid.

Proof of Theorem 1. Lemma 7 proves that by Line 5 of Algorithm 1, the connected components of the matroid $(M, \mathcal{L}, u)$ will be uniform matroids and key matroids.

To prove that the final allocation is EF1, we inductively show that the following hold each time the algorithm reaches Line 7:

- A is EF1, and
- O is the reverse topological envy order of A, i.e., under A, no agent envies another agent who appears earlier in the ordering O.

Clearly, this is true in the beginning when the algorithm first arrives at Line 7 with $A = \emptyset$. Suppose it is true when the algorithm arrives at Line 7 with current allocation A and appends allocation $A^{\mathcal{C}}$ of the next connected component $\mathcal{C}$ to A.

Note that $A^{\mathcal{C}}$ is EF1 and O is its topological envy order. When $\mathcal{C}$ is a uniform matroid, this is a well-known property of the round-robin method [Caragiannis *et al.*, 2019]. When $\mathcal{C}$ is a key matroid, this is established in Lemma 8. Finally, it is also known in the literature that taking the union of two EF1 allocations whose topological envy orders are the reverse of each other yields an EF1 allocation (e.g., see Biswas and Barman [2018, Lemma 2]). Hence, A is still EF1 at the end of Line 13. Then, after applying envy-cycle elimination in Line 14, A remains EF1 and O is set to be its new reverse topological envy order, proving the induction step.

Finally, it is easy to observe that all our operations, the round-robin method, the method to solve the key matroid from the proof of Lemma 8, the envy-cycle elimination procedure of Lipton *et al.* [2004], and all other steps in Algorithm 1 can be executed in polynomial time. Hence, Algorithm 1 runs in polynomial time. □

Algorithm 1: 3-Agent EF1 for Laminar Matroids

Input: Additive valuations (v_1, v_2, v_3), laminar matroid $(M, \mathcal{L}, u)$

1 $(M, \mathcal{L}, u) \leftarrow$ Apply laminar packing to $(M, \mathcal{L}, u)$
2 **while** *any operation* $\mathrm{op} \in \{$*splitting, child pruning, parent pruning, partition extraction*$\}$ *can be applied to* $(M, \mathcal{L}, u)$ **do**
3     $(M, \mathcal{L}, u) \leftarrow$ Apply op to $(M, \mathcal{L}, u)$
4 **end**
5 $O \leftarrow (1, 2, 3)$
6 $A \leftarrow (A_1 = \emptyset,\ A_2 = \emptyset,\ A_3 = \emptyset)$
7 **for** *each connected component* $\mathcal{C}$ *of* $(M, \mathcal{L}, u)$ **do**
8   **if** $\mathcal{C}$ *is a uniform matroid* **then**
9         $A^{\mathcal{C}} \leftarrow$ Allocate the goods in $\mathcal{C}$ via the round-robin method with picking order O
10   **else**
11         $A^{\mathcal{C}} \leftarrow$ Allocate the goods in $\mathcal{C}$ via Lemma 8 with the no-envy order O
12   **end**
13     $A \leftarrow A \cup A^{\mathcal{C}}$
14     $A \leftarrow$ Apply envy-cycle elimination of Lipton *et al.* [2004] to A
15     $O \leftarrow$ reverse topological order of the agents from the envy graph (so no agent envies any agent earlier in the ordering O)
16 **end**
17 **return** A

4 Matroids With More Agents

We achieve our main result by decomposing any laminar matroid into a new matroid that is *almost* a partition matroid. If there was a way to continue the decomposition process, breaking the key matroids down into uniform matroids as well, then we would no longer need our additional proofs on how to solve key matroids. We could simply use the algorithm of Biswas and Barman [2018] without any modifications on the fully decomposed matroid to get a solution. However, it is not hard to confirm that for the key matroid $(M, \mathcal{L}, u)$, there does not exist any 3-colorable partition matroid $(M, \mathcal{I})$ such that $\mathcal{I} \subseteq \mathcal{I}(M, \mathcal{L}, u)$, so there is no way to continue this decomposition process while maintaining 3-colorability and the Independent Set Shrinkage property, which are both vital for our proof.

Interestingly, the work of Bérczi *et al.* [2021] also studies the technique of taking a complex matroid, and reducing it to a partition matroid that meets the Independent Set Shrinkage property, then solving their problem on that simpler partition matroid (the problem they study is list-coloring two matroids). However, their reduction is different in a key way. They allow the final partition matroid to be more than n-colorable by up to a factor of 2. With this relaxation, they are able to reduce their matroids to *exactly* partition matroids, without any pesky key matroid-like pieces left lying around. Below we state their result formally.

Theorem 2. *For any n-colorable matroid $(M, \mathcal{I})$, there exists a $f(n)$-colorable partition matroid $(M, \mathcal{I}')$ such that $\mathcal{I}' \subseteq \mathcal{I}$, where $f(n)$ depends on the class of $(M, \mathcal{I})$ as follows:*

1. *If $(M, \mathcal{I})$ is a transversal matroid, then $f(n) = n$ (Theorem 1.5 of [Bérczi et al., 2021]).*
2. *If $(M, \mathcal{I})$ is a graphic matroid, then $f(n) = 2n - 1$ (Theorem 1.6 of [Bérczi et al., 2021]).*
3. *If $(M, \mathcal{I})$ is a gammoid, then $f(n) = 2n - 2$ (Theorem 1.7 of [Bérczi et al., 2021]).*

It turns out that this style of reduction plus the partition matroid algorithm of Biswas and Barman [2018] also yields some very interesting results, which we illustrate below:

Lemma 9. *Let $(M, \mathcal{I})$ be a matroid and let $(M, \mathcal{I}')$ be a k-colorable partition matroid with the property that $\mathcal{I}' \subseteq \mathcal{I}$. Then, there exists an EF1 allocation on M to k agents that is feasible with respect to $(M, \mathcal{I})$.*

Proof. The algorithm of Biswas and Barman [2018] can be used to find an EF1 allocation A of M to k agents that is feasible with respect to $\mathcal{I}'$. By definition, we must have that $\mathcal{I}' \subseteq \mathcal{I}$, so A will be feasible with respect to $\mathcal{I}$ as well. □

Combining Theorem 2 with Lemma 9, we obtain the following result:

Theorem 3. *For any n-colorable matroid $(M, \mathcal{I})$, there exists a feasible EF1 allocation of M to $f(n)$ agents, where $f(n)$ depends on the class of $(M, \mathcal{I})$ as follows:*

1. *If $(M, \mathcal{I})$ is a transversal matroid, then $f(n) = n$.*
2. *If $(M, \mathcal{I})$ is a graphic matroid, then $f(n) = 2n - 1$.*
3. *If $(M, \mathcal{I})$ is a gammoid, then $f(n) = 2n - 2$.*

Proof. Given an n-colorable matroid $(M, \mathcal{I})$ from any of the above classes, Theorem 2 states that for the appropriate $f(n)$, there exists a $f(n)$-colorable partition matroid $(M, \mathcal{I}')$ such that $\mathcal{I}' \subseteq \mathcal{I}$. Therefore, by Lemma 9, there exists a complete and feasible EF1 allocation of that partition matroid to $f(n)$ agents; thus, that allocation is also feasible for $(M, \mathcal{I})$. □

Most notably, for n-colorable transversal matroids, Bérczi *et al.* [2021] provide a perfect reduction to n-colorable partition matroids. Thus, if a transversal matroid is n-colorable, there always exists a feasible, complete EF1 allocation of the goods in that matroid to n agents.

Transversal matroids are able to model constrained fair division instances that require role fulfillment. A prime example is the fair allocation of individuals into balanced teams from a diverse talent pool. One can imagine dividing athletes into multiple sports teams: each athlete can play a specific set of positions, and every team must be constructed so that all required positions are filled by exactly one capable player. These perfectly formed teams represent the "bases" of a transversal matroid. Thus, using this algorithm, we would always be able to construct teams such that the coach of each team gets a collection of players they view as fair, while ensuring that all the teams are strictly functional on the field.

For graphic matroids and gammoids, Theorem 3 represents proving a weaker version of the EF1 matroid conjecture for these classes. The full conjecture states that if a matroid from these classes admits a feasible allocation to n agents, then it will admit an EF1 feasible allocation to n agents. Theorem 3 retains the hypothesis, but only allows us to conclude that a feasible EF1 allocation exists for $f(n) > n$ agents. Fully resolving the conjecture for these classes would involve proving EF1 existence for all numbers of agents between n and $2n - 1$ or $2n - 2$ respectively.

5 Discussion

EF1 with ordered no-envy. Our work suggests that the general approach of simplifying a given matroid into small connected components via feasibility-preserving operations and then allocating the connected components sequentially is promising for matroid-constrained fair division. However, this requires finding an allocation of each connected component that is not only EF1, but also has a directed acyclic envy graph with a given topological order. Whether such allocations exist subject to all matroid constraints (that admit a feasible allocation) is a stronger question than that of EF1 existence. Such open questions have also been posed when seeking additional properties such as Pareto optimality [Freeman *et al.*, 2020].

Resolving the case of four agents. In Appendix B, we show that our technique reduces the existence of EF1 allocations to four agents under laminar matroids to the existence of EF1 allocations with the ordered no-envy condition in two small laminar matroids. Our efforts for a computer-aided resolution of this missing piece remained unfruitful, but this result may be within reach with a modest amount of effort, possibly even with a smarter brute-force strategy.

Matroid decompositions. In Section 4, we use partition decomposition bounds of Bérczi *et al.* [2021] to show EF1 existence under various classes of matroids subject to stronger-than-feasibility conditions. Bérczi *et al.* [2021] show that their bounds are tight, so improving them is not a viable approach to strengthening these results. However, our approach of decomposing into a partition matroid along with small other matroids seems to be a promising future avenue. In particular, can we design feasibility-preserving operations for broader classes of matroids than laminar matroids?

Acknowledgements

The authors were supported by an NSERC Discovery Grant and an NSERC-CSE Research Communities Grant. Researchers funded through the NSERC-CSE Research Communities Grants do not represent the Communications Security Establishment Canada or the Government of Canada. Any research, opinions or positions they produce as part of this initiative do not represent the official views of the Government of Canada.

References

Hannaneh Akrami and Nidhi Rathi. Achieving maximin share and efx/ef1 guarantees simultaneously. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, pages 13529–13537, 2025.

Hannaneh Akrami, Siyue Liu, Roshan Raj, and László A Végh. Matroids are equitable. *arXiv preprint arXiv:2507.12100*, 2025.

Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi-Agent Systems*, 36(1):3, 2022.

Haris Aziz, Rupert Freeman, Nisarg Shah, and Rohit Vaish. Best of both worlds: Ex ante and ex post fairness in resource allocation. *Operations Research*, 72(4):1674–1688, 2023.

Kristóf Bérczi, Tamás Schwarcz, and Yutaro Yamaguchi. List coloring of two matroids through reduction to partition matroids. *SIAM Journal on Discrete Mathematics*, 35(3):2192–2209, 2021.

Péter Biró, Gergely Csáji, and Ildikó Schlotter. Stable hypergraph matching in unimodular hypergraphs. *arXiv preprint arXiv:2502.08827*, 2025.

Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 91–97, 2018.

Arpita Biswas and Siddharth Barman. Matroid constrained fair allocation problem. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 9921–9922, 2019.

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

Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation*, 7(3): Article 12, 2019.

Benjamin Cookson, Soroush Ebadian, and Nisarg Shah. Constrained fair and efficient allocations. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, pages 13718–13726, 2025.

Dominique de Werra. Equitable colorations of graphs. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 5(R3):3–8, 1971.

Amitay Dror, Michal Feldman, and Erel Segal-Halevi. On fair division under heterogeneous matroid constraints. *Journal of Artificial Intelligence Research*, 76:567–611, 2023.

Sarfaraz Equbal. Fair division under laminar matroid constraints for three agents. In *Proceedings of the 25th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2026. Forthcoming.

Rupert Freeman, Nisarg Shah, and Rohit Vaish. Best of both worlds: Ex-ante and ex-post fairness in resource allocation. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 21–22, 2020.

Harold Gabow. Decomposing symmetric exchanges in matroid bases. *Mathematical Programming*, 10(1):271–276, 1976.

Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 6th ACM Conference on Economics and Computation (EC)*, pages 125–131, 2004.

Hila Shoshan, Erel Segal-Halevi, and Noam Hazon. Efficient nearly-fair division with capacity constraints. *arXiv preprint arXiv:2205.07779*, 2022.

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

Appendix

A Omitted Proofs from Section 3

A.1 Omitted Proofs from Section 3.1

Lemma 10. *A laminar matroid $(M, \mathcal{L}, u)$ is n-colorable if and only if $u(S) \geqslant \lceil |S|/n \rceil$ for all $S \in \mathcal{L}$.*

Proof. Necessity is simple. Clearly to partition the elements of a set S into n disjoint subsets, some subset must contain at least $\lceil |S|/n \rceil$ elements. Thus, if $u(S) < \lceil |S|/n \rceil$, there is no way to partition the full set of items M into n disjoint subsets in a way that meets this constraint.

To prove sufficiency, we appeal to a number of different theorems. First, it is well known that when a laminar set family is represented as a hypergraph (each element is a vertex and the edges are the sets in the family), then the incidence matrix of that hypergraph will be totally unimodular (see for example [Biró *et al.*, 2025]). Then, from [de Werra, 1971], we know that for any hypergraph with such a property, and any $k \geqslant 1$, it is possible to partition the vertices of the hypergraph into k disjoint sets, such that no edge S of the hypergraph has more than $\lceil |S|/k \rceil$ vertices in the same set. We can set $k = n$ and therefore get the existence of a valid n-coloring of our laminar matroid. □

B Four Agents

In this section, we show which parts of our techniques can and cannot be applied to the case $n = 4$ agents under laminar matroid constraints, both highlighting the versatility of our decomposition techniques, and showing why EF1 under laminar matroid constraints beyond 3 agents is a difficult problem. Recall that the packing and simplification steps in Sections 3.1 and 3.2, respectively, hold for any number of agents. Only the steps to decompose into uniform and key matroids in Section 3.3 and solve the key matroid in Section 3.4 are specific to the case of $n = 3$. We show that similar steps can be executed for higher n, although it starts to get significantly trickier.

In Section 3.3, we introduced the *key matroid*, which was integral to our decomposition process for 3-colorable matroids. In the case of 4-colorable matroids, we take a similar approach. However, in this case, we must introduce two key matroids rather than one.

Definition 13 (Size-8 Key Matroid)**.** We say that a laminar matroid $(M, \mathcal{L}, u)$ is a *size-8 key matroid* if:

- $|M| = 8$,
- $|\mathcal{L}| = 4$,
- $M \in \mathcal{L}$ with $u(M) = 2$, and
- the remaining three sets in $\mathcal{L}$ form a disjoint partition of M with sizes (3, 3, 2), and each have an upper bound of 1.

Definition 14 (Size-12 Key Matroid)**.** We say that a laminar matroid $(M, \mathcal{L}, u)$ is a *size-12 key matroid* if:

- $|M| = 12$,
- $|\mathcal{L}| = 5$,
- $M \in \mathcal{L}$ with $u(M) = 3$, and
- the remaining four sets in $\mathcal{L}$ are all size 3, form a disjoint partition of M, and each have an upper bound of 1.

Using these two key matroids, we will prove an analogous result to the 3-colorable matroid decomposition.

Lemma 11. *Let $(M, \mathcal{L}, u)$ be a laminar matroid that is 4-colorable and packed. If none of splitting, child pruning, parent pruning, and partition extraction operations can be applied to $(M, \mathcal{L}, u)$, then every connected component of $(M, \mathcal{L}, u)$ is either a uniform matroid or a key matroid.*

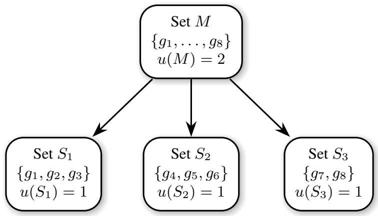

Figure 2: Size 8 Key Matroid

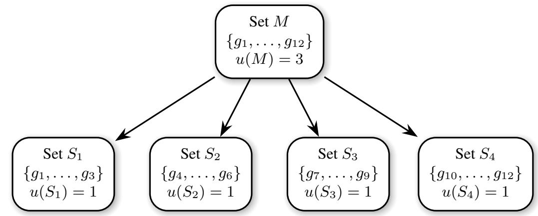

Figure 3: Size 12 Key Matroid

Figure 4: Visualizations of the Size 8 and Size 12 Key Matroids.

Proof. Fix a connected component $(M', \mathcal{L}', u')$ and for the sake of contradiction, assume it is neither uniform nor one of the two key types of 4-colorable matroids; we derive a contradiction. Since the matroid is 4-packed, $u(S) = \lceil |S|/4 \rceil$ for all $S \in \mathcal{L}'$.

First, we will show that every leaf of $(M', \mathcal{L}', u')$ has size at most 3. Let S be a leaf, i.e., $D(S) = \emptyset$. If $|S| = 4$, then *partition extraction* applies. If $|S| > 4$, then *splitting* applies to S because S is a leaf, so the conditions of splitting on $C(S)$ are vacuous, and thus we can let S_1 be any subset of S of size 4. Hence every leaf satisfies $|S| \leqslant 3$.

Select some $S \in \mathcal{L}'$ such that every $C \in C(S)$ is a leaf (such a set must exist, or else $(M', \mathcal{L}', u')$ would be a uniform matroid). $C(S)$ is a disjoint partition of S, and each $C \in C(S)$ has $|C| \leqslant 3$. Also, note that $|S| > 4$, otherwise *child pruning* applies.

Next, we can show that for any $C \in C(S)$, $|C| > 1$ will be true. To see this, note that if there exist two children in $C(S)$ of sizes 1 and 3, their union has size 4, so *splitting* can be applied to S (take S_1 as that union). This same argument can be made if there are at least four size-1 children in $C(S)$ or at least 2 size 2 children in $C(S)$. Therefore if $C(S)$ were to include a set of size 1, it could include no sets of size 3 and at most a single set of size 2. This means that we would have $\sum_{C \in C(S)} |C| \leqslant 3 < 4 < |S|$, which contradicts the fact that $C(S)$ partitions S. Therefore, to avoid splitting, we must have no size-1 children. Also note that using from the logic above, we can also conclude that $C(S)$ contains at most a single set of size 2.

With this information, we will examine exactly what forms the multi-set of sizes of the sets in $C(S)$ can take.

First, note that $|C(S)| > 2$ must be true. This follows from the fact that $(M', \mathcal{L}', u')$ is packed, we must have $u'(S) \geqslant \lceil 5/4 \rceil = 2$, and for each $C \in C(S)$, we have that $u'(C) \leqslant \lceil 3/4 \rceil = 1$. Thus, if $|C(S)| \leqslant 2$, we would have that $u'(S) \geqslant 2 \geqslant \sum_{C \in C(S)} u'(C)$, and the parent pruning operation would apply.

Since $|C(S)| > 2$, and we know that there can be no sets of size 1, and at most a single set of size 2, this means there must be at least 2 sets of size 3.

Let $k_3 \geqslant 2$ be the number of size-3 sets in $C(S)$ and $k_2 \in \{0, 1\}$ indicate whether there is a size-2 set in $C(S)$. Then $|S| = 3k_3 + 2k_2$.

Case A: $k_2 = 0$ (all children have size 3). If $k_2 = 0$, then by the fact that $|C(S)| \geqslant 3$, we must have $k_3 \geqslant 3$. If $k_3 = 3$, then $|S| = 9$ and $u'(S) = \lceil 9/4 \rceil = 3$, while $\sum_{C \in C(S)} u'(C) = 3$, so *parent pruning* would apply. If $k_3 \geqslant 5$, then the union of any four children has size 12 (a positive multiple of 4) and is a proper subset of S, so *splitting* applies. Hence the only possibility is $k_3 = 4$, i.e., $|S| = 12$ with four size-3 children.

Case B: $k_2 = 1$ (one size-2 child and the rest size 3). If $k_3 \geqslant 3$, then two size-3 children plus the size-2 child have union size 8 (a positive multiple of 4), which is a proper subset of S, so *splitting* applies. Hence the only possibility is $k_3 = 2$, i.e., $|S| = 8$ with child sizes $(3, 3, 2)$.

Thus S is either (i) size 12 with four children of size 3, or (ii) size 8 with children of sizes $(3, 3, 2)$.

If there are no other sets in this component, then S and its children would form one of the two key matroids. Therefore, there must be some other set in $\mathcal{L}'$. Specifically, there must be some $T \supset S$. However, note that in either of our cases, S is size 12 or 8, both of which are multiples of 4. Therefore, we would be able to perform splitting on T, taking $S_1 = S$. Therefore S is top-level in this component, so the component is either the size-8 or size-12 key matroid, a contradiction. □

Using Lemma 11, we can decompose 4-colorable laminar matroids in the same way that we decomposed 3-colorable ones. If we could find ways to "solve" the two 4-colorable key matroids (i.e., find an EF1 allocation over the goods to 4 agents that also has the "no ordered envy" property), then we could combine these results with Algorithm 1 to show the existence of EF1 allocations for all 4-colorable laminar matroids to 4 agents. In the below lemma, we show that this is not as straightforward as in the 3 agent case, as the technique we use to "solve" the 3-colorable key matroid no longer works.

Lemma 12. *It is not always possible to construct an allocation over the goods in the size-8 laminar matroid to 4 agents such that for each agent $i \in N$, that agent's top i favorite goods all appear in different bundles.*

Proof. By the constraints imposed by the size-8 key matroid, we must have that:

- $|A_i| = 2$ for all $i \in N$.
- For any set $S \in \{\{g_1, g_2, g_3\}, \{g_4, g_5, g_6\}, \{g_7, g_8\}\}$, no more than a single good from S can appear in any bundle.

Assume that agent 4's top 4 favorite goods are $\{g_1, g_2, g_5, g_6\}$ and agent 3's top 3 favorite goods are $\{g_3, g_5, g_6\}$ (The top goods of agents 1 and 2 are arbitrary).

From the constraint induced by agent 4's top 4 favorite goods, we know that in any valid allocation, each bundle contains exactly one of the goods in $\{g_1, g_2, g_5, g_6\}$. From the constraints of the key matroid, g_3 cannot appear in the same bundle as g_1 or g_2. Further, from the constraint induced by agent 3's top 3 favorite goods, g_3 cannot appear in the same bundle as g_5 or g_6. However, this means that g_3 cannot appear in any bundle. □

This shows that we cannot use the same proof technique we used in the 3-colorable case in order to solve these larger key matroids, but it still leaves open the question of whether an allocation with the desired properties always exists.