EFX Allocation in (Multi)Hypergraphs

Thanasis Lianeas[1] **Alkmini Sgouritsa**[2,3] **Minas Marios Sotiriou**[2,3]

[1]University of West Attica
[2]Athens University of Economics and Business
[3]Archimedes/Athena RC
lianeas@corelab.ntua.gr, alkmini@aueb.gr, minas_marios@outlook.com

Abstract

We study fair allocations of *indivisible* goods among agents with heterogeneous monotone valuations. As fair we consider the allocations that are envy-free-up-to-any-good (EFX). Finding if EFX allocations always exist, even for agents with additive valuations, is a major open problem in Fair Division. Christodoulou et al. (2023) introduced the (multi-hyper)graph setting, where agents and goods are represented by vertices and edges of a graph, respectively, and only the endpoints of an edge may have non-zero marginal value for it. We show that for *hypergraphs* with girth at least 4 and agents with *general monotone* valuations there always exists an EFX allocation and can be constructed in polynomial time. We generalize our approach to also show that multi-hypergraphs with girth (on the simple hypergraph) at least 4 always admit an EFX allocation, as long as there exists a single vertex whose incident edges have multiplicity at most the size of that edge minus 2; our construction in this case needs pseudo-polynomial time.

1 Introduction

Fair division of resources among agents is an important topic, from dividing inheritance, to allocating computational resources to training models. Many situations fall under the category of fair division of indivisible goods, which is the central theme of this paper. There are sites (e.g. http://www.spliddit.org/) that provide mechanisms for such applications, based on theoretical results. The research of fair division dates back to almost 80 years ago (Steihaus 1948). A well studied and established notion of fair allocations are *envy-free allocations*, where nobody envies another agent (Gamow and Stern 1958; Foley 1966; Varian 1974), which always exists for divisible resources (Stromquist 1980; Woodall 1980; Aziz and Mackenzie 2016). However, envy-free allocations may not always exist regarding indivisible goods; consider for instance the case of two agents and a valued good, where whoever gets the good is envied by the other agent. Since the envy-free condition is too strict in this case, two natural relaxations were defined. The first notion is envy-freeness up to one good (EF1) (Budish 2010). Such an allocation always exists and can be computed in polynomial time (Lipton et al. 2004), even when agents have arbitrarily heterogeneous (monotone) valuations over the sets of goods. The second notion, which is the one considered here, is envy-freeness up to any good (EFX) (Caragiannis et al. 2019) and it is stricter than EF1.

In contrast to EF1, it is unknown if EFX allocations always exist, and this problem has been described as "Fair Division's Most Enigmatic Question" (Procaccia 2020). An EFX allocation is known to exist only in special cases: for 2 agents with heterogeneous monotone valuations (Plaut and Roughgarden 2020), for 3 agents with additive valuations or a slightly more general class of valuations (Chaudhury, Garg, and Mehlhorn 2024; Akrami et al. 2023), and for many agents with identical monotone valuations (Plaut and Roughgarden 2020). Other results limit the valuation function of the agents or consider limited types of agents (e.g., (Amanatidis et al. 2021; Hv et al. 2025; Hosseini et al. 2021; Babaioff, Ezra, and Feige 2021)).

Christodoulou et al. (2023) introduced a restriction on the valuations by relying on a graphical structure. More precisely, agents and edges are represented by vertices and edges, respectively, and *only* the agents that are the endpoints of an edge/good may consider it valuable. The motivation behind the graph setting is about instances where multiple agents find valuable a "neighboring" good but not all of them (e.g. geographic settings between neighboring countries, or allocation of work space between research labs). Moreover, the multi-hypergraph setting is basically the unrestricted setting, therefore, exploring ways of coping with graph settings may be proven useful for solving the general problem. Christodoulou et al. (2023) showed that it is not always possible to construct an EFX allocation by orienting the edges of the graph, i.e., by allocating each edge to one of its endpoints, even for 4 agents, however, they showed that EFX allocations always exist.

There have been several follow-up works considering the existence of EFX allocations in graph settings: Kaviani, Seddighin, and Shahrezaei (2024) showed existence of EFX allocations for the multigraph setting when agents have restricted additive valuations,[1] and there is a line of works showing existence of EFX allocations in multigraphs for

[1]In restrictive additive valuations each agent values each good g by either a fixed value v_g for good g, or 0. Then, the valuation for any set equals the sum of the values for each good in the set.

more general valuations under restrictions on the graph structure (Afshinmehr et al. 2025; Bhaskar and Pandit 2024; Sgouritsa and Sotiriou 2025); a common restriction that appears in all those works is about the girth of the underlying simple graph. Our results is in the same direction and focus on the existence of EFX allocations in hypergraph and multi-hypergraph setting with girth at least 4. We note that for hypergraphs with girth at least 3, the best known result is only an approximation of the EFX guarantee ($\frac{\sqrt{2}}{2}-$EFX existence) for subadditive valuations (Kaviani, Seddighin, and Shahrezaei 2024); we manage to show exact EFX guarantees for *general* monotone valuations by slightly relaxing the restriction on the girth.

Our Results. We show that an EFX allocation always exists for hypergraphs of girth at least 4, when agents have general heterogeneous monotone valuations. We do this by constructing this allocation in polynomial time. The following theorem describes this first result.

Theorem 1. *Instances on hypergraphs of girth at least 4 always admit an EFX allocation that can be constructed in polynomial time to the number of agents and goods.*

We generalize our result, by considering a multi-hypergraph of girth at least 4: such a graph may contain multiple edges of the same subset of vertices, i.e., each edge may have multiplicity more than 1, but the simple underlying hypergraph (by considering multiplicity 1 for all edges) has girth at least 4. We remark that with no further restriction, the existence of EFX allocations in such a setting reduces to the general problem of EFX existence: consider a single edge containing all vertices of multiplicity equal to the number of goods. In our second result we prove the existence of EFX allocation in multi-hypergraphs of girth at least 4 as long as there exists a single vertex whose incident edges have multiplicity at most the size of that edge minus 2; our construction in this case needs pseudo-polynomial time. We justify the importance of this restriction by showing that if we slightly relax this restriction, the problem of EFX existence with at most one unallocated good reduces to the problem we try to solve; this problem is considered quite difficult and it is solved only for the case of 4 agents (Berger et al. 2022). Our second result is summarized in the next theorem.

Theorem 2. *For multi-hypergraphs with girth at least 4 where there exists a vertex whose incident edges have multiplicity at most the edge's size minus 2, there exists an EFX allocation constructed in pseudo-polynomial time.*

Our approach. It is well known that EFX orientations may not exist even in simple graphs (see the counterexample in (Christodoulou et al. 2023) regarding the existence of EFX orientations), hence, it may be the case that in an EFX allocation some edges are allocated to vertices that are not endpoints of those edges. Our approach anticipates this fact and initially chooses an arbitrary vertex (vertex 0 in our analysis) to serve as the vertex to "park" those edges. One such vertex is sufficient in the case of simple hypergraphs (Theorem 1), however when we consider edges of higher multiplicity than 1 (Theorem 2), more such vertices may be needed, and for that we employ the neighbors of vertex 0 to play such a role.

Further Related Work. We focus on references related to multigraphs and hypergraphs.

Christodoulou et al. (2023) introduced the graph setting and showed the existence of EFX allocation when the graph is simple and the items are goods. This result was extended to the case of mixed manna, where items may be both goods and chores (Zhou et al. 2024). Kaviani, Seddighin, and Shahrezaei (2024) showed that multigraphs where agents have restricted additive valuations always admit an EFX allocation. Afshinmehr et al. (2025), and Bhaskar and Pandit (2024) proved that bipartite multigraphs where agents have additive and cancelable valuations, respectively, admit EFX allocations. Also Bhaskar and Pandit (2024) showed that t colored multigraphs with girth at least $2t-1$ for agents with cancelable valuations admit EFX allocations, and Sgouritsa and Sotiriou (2025) showed that multigraphs with girth at least 6 or multigraphs where each vertex has at most $\lceil \frac{n}{4} \rceil - 1$ neighbors admit EFX allocations for agents with general monotone valuations.

Approximate EFX-allocations have also been studied in multigraphs and hypergraphs. Amanatidis, Filos-Ratsikas, and Sgouritsa (2024) showed that $\frac{2}{3}-$EFX allocations always exist in multigraphs when agents have additive valuations, which was recently improved to a $\frac{\sqrt{2}}{2}-$EFX (Kaviani et al. 2025). Kaviani, Seddighin, and Shahrezaei (2024) showed that for hypergraphs with girth at least 3 (meaning that any two agents share at most one edge), $\frac{\sqrt{2}}{2}$-EFX allocations always exist when agents have subadditive valuations.

In the graph and multigraph setting the existence of EFX orientations, i.e., allocations where edges may only be allocated to one of the endpoints, has also been considered. Christodoulou et al. (2023) showed that EFX orientations need not exist by giving a counterexample in a K_4 graph, and they further showed that even deciding if an EFX orientation exists is NP-complete; it was later shown that this result holds even if the vertex cover of the graph has size 8, or in multigraphs with only 10 vertices (Deligkas et al. 2024). Zeng and Mehta (2024) showed that EFX orientations may not exist in graphs with chromatic number greater than 3, and they always exist when the chromatic number is at most 2. The complexity of orientations has been also explored in multigraphs (Hsu 2024; Afshinmehr et al. 2025).

Another fairness criterion known as maximin share (MMS) has also been explored in multigraphs (Christodoulou and Mastrakoulis 2025; Feige 2025).

2 Preliminaries

We consider a setting where there is a set N of n agents and a set M of m indivisible goods, and each agent i has a valuation function $v_i : 2^M \to \mathbb{R}$, over the subsets of goods, i.e., $v_i(S)$ denotes the valuation of agent i for the subset S of goods. The valuation functions are considered to be monotone, i.e., for any $S \subseteq T \subseteq M$, it holds that $v_i(S) \leq v_i(T)$, and normalized, i.e., $v_i(\emptyset) = 0$. For simplicity, for the valuation of i for some good g, we write $v_i(g)$ instead of $v_i(\{g\})$. An *allocation* $\mathbf{X} = (X_1, \ldots, X_n)$ is a partition

of a subset of goods into n disjoint bundles $X_1, \ldots, X_n$, where each agent i receives X_i. An allocation is *complete* if it allocates all the goods, and *partial* if not. Throughout, for $k \in \mathbb{N}^+$, we let $[k] := \{1, 2, \ldots, k\}$.

Envy - EFX allocation. Given an allocation $\mathbf{X} = (X_1, \ldots, X_n)$, an agent i *envies* an agent j (or alternatevly X_j), if $v_i(X_i) < v_i(X_j)$. An allocation $\mathbf{X} = (X_1, \ldots, X_n)$ is *EFX* if for any pair of agents i, j it holds that:

$$v_i(X_i) \geq v_i(X_j \setminus \{g\}) \ , \forall g \in X_j.$$

In other words, in an EFX allocation, no agent envies a proper subset of what is allocated to any other agent.

We assume that the setting is modeled on hypergraphs. A *hypergraph* is a pair $G = (V, E)$ where V is a set of vertices and E is a set of subsets of V, called hyperedges or simply edges. The size of an edge is the number of vertices it contains. For an edge $e \in E$ and a vertex $i \in e$ we say that e is incident to i and i is incident to e, or i is an endpoint of e. If vertices i and j belong to some edge e (i.e., $i, j \in e$), we say that i and j share e and we call i and j neighbors.

Cycles (Berge 1973) - Girth. For a hypergraph G, a cycle of length k is defined by a sequence $(x_1, e_1, x_2, e_2, \ldots, x_k, e_k, x_{k+1})$ such that: (i) $x_1, x_2, \ldots, x_k$ are distinct vertices and $x_{k+1} = x_1$, (ii) $e_1, e_2, \ldots, e_k$ are distinct edges and (iii) $x_i, x_{i+1} \in e_i$, $\forall i \in [k]$. A hypergraph G has *girth* k, if G's shortest cycle has length k; if there is no cycle in G the girth is infinity.

Hypergraph setting. In the hypergraph setting, we let agents correspond to vertices of a hypergraph $G = (V, E)$, and goods correspond to edges of G. For ease, we refer to the agents as vertices and to goods as edges. The edges that are not incident to some vertex i are *irrelevant* to it, i.e., for any $S \subseteq E$ and any $e \in E$ such that $i \notin e$, $v_i(S \cup \{e\}) = v_i(S)$. Note that this implies that for $e \in E$ irrelevant to $i \in V$: $v_i(e) = v_i(\emptyset) = 0$. We call an edge e *relevant* to i, if $i \in e$.

Orientation. An allocation $\mathbf{X}$ is an *orientation* if for any allocated edge e, if $e \in X_i$, then i is an endpoint of e.

Unallocated edges. Given a partial allocation $\mathbf{X}$, an edge e is *unallocated* if $e \notin X_i$, for all $i \in V$. We denote by $U(\mathbf{X})$ the set of unallocated edges in $\mathbf{X}$. For each vertex i, we define $U_i(\mathbf{X})$ to be the set of all the unallocated edges that are relevant to i.

We give the following two observations for hypergraphs with girth at least 3 and at least 4. For hypergraphs with girth at least 3, any two vertices may share at most one edge, and for hypergraphs with girth at least 4, any two vertices of an edge e do not share any other common neighbor outside e.

Observation 2.1. *In hypergraphs with girth at least 3, any two vertices may share at most one edge.*

Proof. On the contrary, let i, j be two vertices that both belong to edges, say, e_1 and e_2. Starting with vertex i, following edge e_1 to reach j and then following edge e_2 to reach i we get a cycle of length 2, i.e., the cycle (i, e_1, j, e_2, i), contradicting the hypothesis that the girth is at least 3. □

Observation 2.2. *In hypergraphs with girth at least 4, any two vertices of an edge e do not share any other common neighbor outside e.*

Proof. On the contrary, let vertices x, y belong in some edge e and let v be one of their common neighbors outside e. Let e_x be an edge that x and v share. By Observation 2.1 y cannot belong in e_x, or else x and y would share e and e_x. Let e_y be an edge that y and v share. The cycle $(x, e_x, v, e_y, y, e, x)$ has length 3, which is a contradiction. □

Multi-hypergraph setting. In Section 4 we allow for a more general hypergraph setting. In the *multi-hypergraph setting* the edge set E is allowed to have edges of the same set of vertices. Such edges correspond to different goods, with possibly different impact on the valuation functions of the vertices/agents. For an edge e that appears k times in E we say that e has multiplicity k. For a multi-hypergraph $G = (V, E)$ we define the girth of G to be the girth of $G' = (V, E')$, where E' is derived from E if we delete all repetitions of the edges of E.

3 EFX on Hypergraphs

For ease of presentation we will rename/reorder the vertices using numbers from 0 to $n - 1$. Pick an arbitrary vertex and let it be vertex 0. Let n_0 be the number of neighbors that vertex 0 has. Arbitrarily, name the neighbors of 0 with numbers from 1 to n_0 and the remaining vertices with numbers from $n_0 + 1$ to $n - 1$. Throughout the runs of the presented algorithms, vertices will be prioritized based on their names-labels, in a decreasing order.

We next repeat the main theorem of the section whose proof is built up by the use of several lemmas.

Theorem 1. *Instances on hypergraphs of girth at least 4 always admit an EFX allocation that can be constructed in polynomial time to the number of agents and goods.*

The proof is constructive and is described in the following subsections. At each time we preserve an EFX allocation by satisfying additional properties. Our intermediate goal, achieved by Algorithm 2, is to achieve a partial allocation by orienting edges that satisfies the following four Properties.

1. $\mathbf{X}$ is a (partial) EFX orientation.
2. Vertex 0 is non-envied and any neighbor i of vertex 0 may be envied only if it is allocated the edge that it shares with 0.
3. For any vertex i and $e \in U_i(\mathbf{X}), v_i(X_i) \geq v_i(e)$.
4. For any envied vertex i, $v_i(X_i) \geq v_i\left(U_i(\mathbf{X})\right)$.

Properties (1)-(2) are preserved throughout Algorithm 2, and Algorithm **FixProp3** therein guarantees Property (3). At the end of Algorithm 2, a stronger property for the envied vertices than (3) is satisfied, namely Property (4). In Algorithm 3 and Lemma 3.5 we show how those properties are used in order to construct a complete EFX allocation.

Proof of Theorem 1. For the proof of Theorem 1 it suffices to run in series Algorithms 2 and 3. By Lemmas 3.2, 3.3, 3.4 and 3.5 we show that if the instance is a hypergraph of girth at least 4, the final allocation will be a complete EFX allocation. In Lemma 3.6 we show that those algorithms run in polynomial time, which completes the proof.

Algorithm 1: **FixProp3**

Input: An allocation $\mathbf{X}$ satisfying Property (1)
Output: An allocation $\mathbf{X}$ satisfying Properties (1) and (3).

1: **while** $\exists\, j \in V, e \in U(\mathbf{X})$: $v_j(e) > v_j(X_j)$ **do**
2: Let j be the maximum vertex with this property for some e
3: $X_j \leftarrow \{\arg\max_{e \in U_j(\mathbf{X})} v_j(e)\}$
4: **end while**
5: **return** $\mathbf{X}$

Algorithm 2: Orienting Edges

Input: A hypergraph G of girth at least 4.
Output: An allocation $\mathbf{X}$ satisfying Properties (1)-(4).

1: Let $\mathbf{X}$ be the all empty allocation.
2: **FixProp3(X)**
3: **while** $\exists$ envied $i \in V$: $v_i(U_i(\mathbf{X})) > v_i(X_i)$ **do**
4: $X_i \leftarrow U_i(\mathbf{X})$
5: **FixProp3(X)**
6: **end while**

Orienting edges. Algorithm **FixProp3** is called in order to ensure that no unallocated edge is preferred by any vertex over its bundle.

Lemma 3.1. *When **FixProp3** is called for an EFX orientation, it outputs an EFX orientation satisfying Property* (3)*.*

Proof. First note that **FixProp3** will terminate since the bundles (and thus the values) that a vertex may get are finite, and at every execution of the while-loop, the value of some vertex for the updated allocation strictly increases.

To see that Property (3) is satisfied in the resulting allocation, see that the condition of the while-loop is the negation of Property (3), so **FixProp3** terminates by satisfying it.

Clearly, the allocating step of line 3 orients a single (unallocated) edge to one of its incident vertices. This keeps the allocation an orientation. This allocation further remains an EFX allocation after each execution of the while-loop, since for the new bundle/edge allocated in line 3, removing the single edge it contains will leave it empty and thus non-envied. Also the value of the vertices may only increase during the execution of **FixProp3**; thus, no further envy may appear in the future and the EFX property will not break. □

Algorithm 2 starts by assigning each vertex, in decreasing order, its most valuable unallocated edge (one call of **FixProp3**). It continues by repeatedly offering envied vertices all their incident unallocated edges in place of their current bundle. If an envied vertex prefers all its incident unallocated edges to its currently allocated bundle, it is allocated those edges, it releases its bundle, followed by one call of **FixProp3**, i.e., repeatedly, any unallocated single edge is offered to vertices, in decreasing order, until no vertex prefers an unallocated edge. This algorithm is similar to Algorithm 2 in (Christodoulou et al. 2023), but edges might have size greater that 2, and an order is added when offering a single unallocated edge, to make sure that Property (2) is satisfied.

Lemma 3.2. *After the termination of Algorithm 2, Properties* (1)*,* (3) *and* (4) *are satisfied.*

Proof. Algorithm 2 will terminate since both **FixProp3** and the allocating step of line 4 only strictly increase the value that a vertex gets and the bundles (and thus the values) that a vertex may get are finite. The condition of the while-loop of Algorithm 2 is the negation of Property (4). Since the condition of the while-loop must be false for the algorithm to terminate, Property (4) will hold. Additionally, since the algorithm ends with a call of **FixProp3**, by Lemma 3.1, Properties (1) and (3) will also be satisfied as long as Property (1) was satisfied before the call of **FixProp3**.

It remains to show that Property (1) was satisfied before any call of **FixProp3**. The empty allocation clearly satisfies Property (1), so it is satisfied before the call of **FixProp3** at line 2. Suppose that Properties (1) and (3) are satisfied before an execution of any round of the while-loop, which is the case before the first execution of the while-loop. It suffices to show that the allocating step of line 4 does not break the EFX property (it clearly gives an orientation); this would guarantee that Property (1) is satisfied before **FixProp3**, and by Lemma 3.1, Properties (1) and (3) would be indeed satisfied before the next round of the while-loop. Consider a vertex i that will get its bundle changed by the allocating step of line 4. By Observation 2.1, for any other vertex j, $U_i(\mathbf{X})$ contains at most one edge, say e, relevant to j. By Property (3), vertex j does not prefer e, if it exists, i.e., $v_j(X_j) \geq v_j(e) = v_j(U_i(\mathbf{X}))$ and thus j will not envy i. Also the value of the vertices may only increase during the execution of Algorithm 2, and therefore no further envy may appear and the EFX property will not break. □

We will prove that Algorithm 2 will output an allocation also satisfying Property (2) in the next two lemmas.

Lemma 3.3. *Vertex* 0 *is non-envied throughout the execution of Algorithm 2.*

Proof. Initially it is $X_0 = \emptyset$ and thus 0 is non-envied. Any time that **FixProp3** is called, in order for 0 to be considered for a change in its bundle, it must be that all other vertices do not prefer any of the unallocated edges, since all of them have higher index from 0. This directly implies that if 0 is allocated some edge by **FixProp3**, no other vertex will envy it and thus, since the values of the vertices may only increase, 0 remains non-envied. Hence, vertex 0 is never considered in the while-loop of Algorithm 2, and overall, it remains non-envied throughout the execution of Algorithm 2. □

Lemma 3.4. *After the termination of Algorithm 2, any neighbor i of vertex* 0 *may only be envied if it is allocated the edge that it shares with* 0*.*

Proof. Consider any vertex $i \in [n_0]$, i.e., a neighbor of 0, and the bundle X_i allocated to i after the termination of Algorithm 2. If $|X_i| = 0$, obviously i is non-envied. If $|X_i| > 1$, then i is still non-envied due to Property (1): Consider any other vertex j and let e be the edge, if any, that it shares with i; by Observation 2.1, i and j may share at most one edge. If $e \notin X_i$, obviously $v_j(X_i) = 0$, and j doesn't

Algorithm 3: Complete allocation

Input: The allocation $\mathbf{X}$ returned by Algorithm 2.
Output: A complete EFX allocation $\mathbf{X}$.

1: **while** $\exists e \in U(\mathbf{X})$ containing a non-envied vertex j **do**
2: $\quad X_j \leftarrow X_j \cup \{e\}$
3: **end while**
4: $X_0 \leftarrow X_0 \cup U(\mathbf{X})$

envy i. If $e \in X_i$, since $|X_i| > 1$, let $e' \neq e$ be some other edge in X_i. Then, by Property (1) it should be that j does not prefer $X_i \setminus \{e'\} \supseteq \{e\}$ to the bundle allocated to j. Since e is the only valuable edge for j in X_i, j does not envy i.

In the case that $|X_i| = 1$, let e be the edge assigned to i and suppose that e is not the edge that i and 0 share. Since Algorithm 2 only orients edges, e is relevant to i. We first show that e cannot be relevant to any other neighbor of 0, and therefore none of them envy i. Let $j \in [n_0]$ be some neighbor of 0 belonging to the edge that 0 and i share. By Observation 2.1, e is irrelevant to j. Let $j \in [n_0]$ be some neighbor of 0 not belonging to the edge that 0 and i share. Due to Observation 2.2, vertices i, j cannot both belong in the same edge, say e', or else they would have vertex 0 as a common neighbor outside e'. Thus, e is irrelevant to j.

Vertices indexed higher than n_0 did not envy i at the time it received X_i during the execution of Algorithm 2: If $X_i = \{e\}$ was allocated to i in a run of **FixProp3**, vertex i was considered as the highest indexed vertex that preferred e to what it got allocated, implying that all vertices with index higher than i preferred their bundles over every single unallocated edge, so e as well. If X_i was allocated to i in the while-loop of Algorithm 2, then Property (3) (from the previous **FixProp3** run) guarantees that nobody envied i at that point. Note that i remains non-envied until the termination of Algorithm 2, since the vertices' value may only increase. □

Complete EFX allocation. In Algorithm 3 we allocate in two steps the remaining edges (if any) to reach a complete EFX allocation. Unallocated edges incident to a non-envied vertex are oriented towards some non-envied vertex and the rest are allocated to 0.

Lemma 3.5. *If $\mathbf{X}$ is the allocation returned by Algorithm 2, then Algorithm 3 returns a complete EFX allocation.*

Proof. There are two steps in Algorithm 3 for allocating the remaining edges, one in line 2, and the other in line 4. We will show that at each of them no further envy is created. In line 2, each unallocated edge containing a non-envied vertex is oriented towards one of its non-envied vertices. This keeps the allocation EFX. To see this, first note that when an edge e is allocated to a vertex i in this way, only vertices incident to e may envy i, and e will be the only edge that they share with i (Observation 2.1). Since $\mathbf{X}$ is an orientation, this implies that for any of these vertices, say j, $v_j(e) = v_j(X_i \cup \{e\})$. Yet, due to Property (3), j prefers its bundle over e, i.e., $v_j(X_j) \geq v_j(e) = v_j(X_i \cup e)$, and thus it will not envy i.

Note that at the end of the while-loop, the allocation is still an orientation and Properties (1)-(4) are still satisfied. Additionally, the remaining unallocated edges have all their incident vertices envied. For the rest of the proof, $\mathbf{X}$ represents the updated allocation after the while-loop.

It remains to show that allocating the rest of the unallocated edges to 0 will not create any envy towords 0. Since all those edges are relevant only to envied vertices, we will show that any envied vertex i will not envy vertex 0 after the allocation in line 4. If i is a neighbor of vertex 0, due to Property (2), i has received the shared edge with 0, and therefore i has no value for X_0. The same holds if i is not a neighbor of vertex 0, since after the while-loop, the allocation is still an orientation. Therefore, in both cases, due to Property (4), $v_i(X_i) \geq v_i(U_i(\mathbf{X})) = v_i(X_0 \cup U(\mathbf{X}))$, and so i does not envy vertex 0 in the final allocation. □

Poly-time EFX construction. To complete the proof of Theorem 1, we show that the construction of the EFX allocation needs polynomial time.

Lemma 3.6. *The construction of the EFX allocation by running Algorithms 2, and 3 needs polynomial time complexity on the number of edges and vertices.*

Proof. Algorithm 2 uses the subroutine **FixProp3** which runs in time $O(n^4)$: Each while-loop needs at most n^2 checks to find an appropriate vertex (since each vertex has degree at most $n-1$), and if those checks follow the priority of the vertices, no extra time is needed to find the maximum vertex (at line 2). Each vertex may update its allocation, i.e., be considered in the while, at most n times (it has at most n relevant edges and any time it strictly increases its value). So, overall the while-loop may be executed at most n^2 times and each execution needs $O(n^2)$ time.

Regarding Algorithm 2, the vertex picked in the while is changed from envied to non-envied (in line 4). We remark that in **FixProp3**, a non-envied vertex may turn to an envied one, however the value each vertex has for its allocated bundle always strictly increases when it updates her bundle. Therefore, each vertex may turn from non-envied to envied at most n times, since every envied vertex receives a single edge and n is an upper bound of its degree. Therefore, the while-loop may be executed at most n^2 times. Overall, the time complexity of Algorithm 2 is $O(n^6)$.

Algorithm 3 allocates at most m edges, each in time $O(n)$, which is the time needed to identify if there exists a non-envied endpoint. Therefore, the time complexity of Algorithm 3 is $O(nm)$. Hence, the construction of an EFX allocation needs overall polynomial time on n and m. □

4 EFX on Multi-hypergraphs

In this section we generalize our approach so it can be applied to the more general setting of multi-hypergraphs. We next repeat the main theorem of this section whose proof is built up by the use of several lemmas.

Theorem 2. *For multi-hypergraphs with girth at least 4 where there exists a vertex whose incident edges have multiplicity at most the edge's size minus 2, there exists an EFX allocation constructed in pseudo-polynomial time.*

Before proceeding to the proof of Theorem 2 we show the necessity of the additional restriction, in the sense that dropping it makes our problem at least as hard as a difficult problem in the literature. More precisely, we construct an instance of a multi-hypergraph with girth at least 4, where there is no vertex whose all incident edges have multiplicity at most the edge's size minus 2, and there exists a vertex that violates this condition only for one of its incident edges, for which the multiplicity is its size minus 1. We show that the general problem of EFX existence with at most one unallocated good reduces to finding an EFX allocation in that instance. We stress out that the problem of EFX existence with at most one unallocated good is considered a hard problem.

Lemma 4.1. *Consider any instance $\mathcal{I}$ of n agents and $m \geq n$ goods, where agents have arbitrary monotone positive valuations. Then, there exists an instance $\mathcal{I}'$ on a multi-hypergraph with $n+1$ vertices, $m+1$ edges and girth at least 4, where there is no vertex whose all incident edges have multiplicity at most that edge's size minus 2, and there exists a vertex with a single incident edge of multiplicity equal its size minus 1, and an EFX allocation in $\mathcal{I}'$ implies an EFX allocation with at most one unallocated good in $\mathcal{I}$.*

Proof. Let $N = [n] = \{1, \dots, n\}$ and $M = \{e_1, \dots e_m\}$ be the set of agents and goods, respectively, in $\mathcal{I}$. We construct a multi-hypergraph instance $\mathcal{I}'$ with vertex set $N' = N \cup \{0\}$, and edge set $M' = M \cup \{e_0\}$, where $e_0 = \{0, 1\}$, and $e_j = [n]$ for all $e_j \in M$, i.e., each edge apart from e_0 contains all the vertices but 0. Note that in this example there is no cycle (the girth is infinity) and there is no vertex whose incident edges have multiplicity at most that edge's size minus 2 and vertex 0 has a single incident edge of multiplicity equal its size minus 1. For any vertex $i \neq 0$, the valuation function v_i over M, coincide with the valuation that they have in $\mathcal{I}$. The valuation for vertex 1 for any set S containing e_0 is 0, if $S = \{e_0\}$, and greater than $v_1(M)$ (its value for all edges except e_0), otherwise. Vertex 0 has some positive value for e_0.

We argue that in any EFX allocation in $\mathcal{I}'$, vertex 0 is allocated e_0 and at most one other edge. First note that if *only* e_0 was allocated to any vertex $i \neq 0$, i has value 0, and some vertex $j \geq 1$ would be allocated at least 2 edges from M. Since vertex i values positively any of those edges, the EFX condition would be violated for i against j. Therefore, if e_0 was allocated to some vertex $i \neq 0$ in any EFX allocation, vertex i should receive at least one more edge. In that case, vertex 0 would envy vertex i after the removal of that edge (since vertex 0 values positively only e_0), and the EFX condition would again be violated. So, we have established that in any EFX allocation, e_0 should be allocated to vertex 0. If vertex 0 was allocated at least two more edges, after the removal of any of those, vertex 1 would still envy vertex 0, which is again a violation of the EFX condition.

Hence, overall, in any EFX allocation vertex 0 is allocated e_0 and at most one other edge. In any such EFX allocation, the EFX condition is satisfied between the vertices in N, and their allocation gives an EFX allocation of the instance $\mathcal{I}$ with at most one unallocated good, namely the good that is possibly given to vertex 0 apart from e_0. □

We now proceed to the proof of Theorem 2. The proof is constructive and generalizes the EFX construction for simple hypergraphs (Section 3). Similarly, our intermediate goal here is the construction of a (partial) EFX allocation that satisfies four properties. The first two properties are as in Section 3 and the other are more generalized to handle the allowance of repetitions of edges. For that we need the following definition regarding multiple appearances of edges.

Definition 4.2. *For a set E of edges where repetitions are allowed, $\mathbf{P}_e$ is the set of edges containing all the appearances of e in E (i.e., edges with the same incident vertices); we refer to $\mathbf{P}_e$ as the* patch *for e. If some edges of E are allocated by $\mathbf{X}$, $U_e(\mathbf{X})$ denotes the subset of $\mathbf{P}_e$ that contains all the unallocated edges of $\mathbf{P}_e$ under $\mathbf{X}$, i.e. $U_e(\mathbf{X}) = U(\mathbf{X}) \cap \mathbf{P}_e$.*

Below we state the four properties for the multi-hypergraph setting. Property (3) differs from the corresponding property in simple hypergraphs in expressing no envy towards the whole unallocated set of any patch; in simple hypergraphs this was just a single edge. Property (4) is extended to consider all vertices (apart from the special vertex 0) and not only the envied ones as in the case of simple hypergraphs; the reason is that we may not be able to orient all edges incident to non-envied vertices, so we need to guarantee that EFX doesn't break when those are allocated to non-incident vertices.[2] The four properties are used in Algorithm 6 and Lemma 4.5 to construct a complete EFX allocation.

1. $\mathbf{X}$ is a (partial) EFX orientation.
2. Vertex 0 is non-envied and any neighbor i of vertex 0 may be envied only if it is allocated some edge(s) that it shares with 0.
3. For any vertex i and any $e \in U_i(\mathbf{X})$, $v_i(X_i) \geq v_i(U_e(\mathbf{X}))$.
4. For any vertex $i \neq 0$, $v_i(X_i) \geq v_i\left(U_i(\mathbf{X})\right)$.

Proof sketch. Due to space limitation we give a proof sketch for Theorem 2. We keep a similar approach to Section 3: we employ an algorithm that outputs an allocation satisfying the four properties, and an algorithm that completes the EFX allocation. We define the vertices' labels as in Section 3 by setting the special vertex whose incident edges have the restricted multiplicity as vertex 0.

Algorithm 5 that satisfies the four properties is the same with Algorithm 2, by substituting the Algorithm **FixProp3** with the more generalized Algorithm **FixProp3Gen**. **FixProp3Gen** basically applies rule U_1 of (Chaudhury et al. 2021) to each patch separately: It checks if there is a set Z of unallocated edges in a *single* patch such that there exists a vertex k that values it more than its bundle, while for the other vertices the EFX condition is satisfied if k receives Z, i.e., they do not envy any proper subset of Z. As long as such a vertex k and set Z exist, the algorithm allocates Z to i (without breaking EFX), while prioritizing vertices with a

[2] We remark that we could transform Algorithm 2 for simple hypergraphs to satisfy this more extended Property (4), however this was not necessary and so we kept only the necessary steps.

Algorithm 4: **FixProp3Gen**

Input: An allocation $\mathbf{X}$ satisfying Property (1).
Output: An allocation $\mathbf{X}$ satisfying Properties (1) and (3).

1: **while** $\exists\, i \in V, e \in E$: $v_i(X_i) < v_i(U_e(\mathbf{X}))$ **do**
2: $Z =$ Subroutine A ($L = e$, $S = U_e(\mathbf{X})$).
3: $k = \arg\max_j \{v_j(Z) > v_j(X_j)\}$
4: $X_k \leftarrow Z$
5: **end while**
6: **return** $\mathbf{X}$

higher index to preserve Property (2). This Algorithm satisfies Property (3), as long as Property (1) was initially satisfied. Algorithm 5 initially calls **FixProp3Gen** to guarantee Properties (1)-(3), and then each vertex, except vertex 0, is offered its relevant unallocated edges resulting in satisfying Property (4); the Algorithm **FixProp3Gen** is executed any time an offer is accepted so that Property (3) is preserved. Algorithm 5 terminates with all four properties satisfied.

The final allocation (Algorithm 6) differs from the one in Section 3 because now there may be unallocated edges with non-envied endpoints that cannot be oriented. In Section 3, the unallocated edges that could not be oriented were given to vertex 0 without causing envy towards 0, since all vertices that had positive value for those edges either were not neighbors to vertex 0 or they had received the *single* edge shared with vertex 0 (Property (2)); in both cases this meant that they had zero value for X_0. In the multi-hypergraph setting, it may be that for vertices that value positively X_0 there are unallocated edges incident to them that cannot be oriented. In that case, no property guarantees that allocating those edges to vertex 0 would not create envy towards 0. For those vertices, we find alternative vertices to "park" their relevant unallocated edges. Next we describe how we do that.

If $X_0 = \emptyset$ at this phase, we allocate all unallocated edges to vertex 0, and due to Property (4) no envy will be created. If $X_0 \neq \emptyset$, then $X_0 \subseteq \mathbf{P}_e$, for some $e \in E$, as $i = 0$ is not considered in line 3 of Algorithm 5. Since $|\mathbf{P}_e|$ is no more than the size of e minus 2 (by Theorem 2's assumption), there would be two vertices $j_1, j_2 \in e$ that are not allocated any edge from $\mathbf{P}_e$. By Property (2), those two vertices are non-envied, and by Observations 2.1 and 2.2, each of the j_1, j_2 is allocated edges that are irrelevant to any vertex of e and their neighbors. So, allocating $U_i(\mathbf{X})$ to j_1, for any $i \in e$ different from j_1 and 0, and $U_{j_1}(\mathbf{X})$ to j_2, creates no further envy due to Properties (3)-(4). For any other vertex $i \notin e$, X_0 is irrelevant for i, and so allocating $U_i(\mathbf{X})$ to vertex 0 again creates no further envy due to Property (4).

Construction of the EFX allocation. We next give Algorithm **FixProp3Gen**; it uses Subroutine A which is Algorithm 3 from (Chaudhury et al. 2021) and returns a minimal set Z of same patch edges that a vertex prefers to its bundle, meaning that no other vertex envies any proper subset of Z.

Lemma 4.3. *When **FixProp3Gen** is called for an EFX orientation outputs an EFX orientation satisfying Property* (3).

Algorithm 5 extends Property (3) to Property (4) for all vertices but 0, while preserving Properties (1) and (2).

Algorithm 5: Orienting Edges

Input: A multi-hypergraph G of girth at least 4, where edges related to vertex 0 have multiplicity at most that edge's size minus 2.
Output: An allocation $\mathbf{X}$ satisfying Properties (1)-(4).

1: **FixProp3Gen(X)**
2: **while** $\exists$ vertex $i \neq 0$: $v_i(U_i(\mathbf{X})) > v_i(X_i)$ **do**
3: $X_i \leftarrow U_i(\mathbf{X})$
4: **FixProp3Gen(X)**
5: **end while**

Algorithm 6: Complete allocation

Input: The allocation $\mathbf{X}$ returned by Algorithm 5.
Output: A complete EFX allocation $\mathbf{X}$.

1: **if** $X_0 \neq \emptyset$ **then**
2: Let e be the edge such that $X_0 \subseteq \mathbf{P}_e$.
3: Let $j_1, j_2 \in e$ be two non-envied vertices where $X_j \cap \mathbf{P}_e = \emptyset$, for $j \in \{j_1, j_2\}$.
4: **for** every $i \in e \setminus \{j_1, 0\}$ **do**
5: $X_{j_1} \leftarrow X_{j_1} \cup U_i(\mathbf{X})$
6: **end for**
7: $X_{j_2} \leftarrow X_{j_2} \cup U_{j_1}(\mathbf{X})$
8: **end if**
9: $X_0 \leftarrow X_0 \cup U(\mathbf{X})$

Lemma 4.4. *After the termination of Algorithm 5, Properties* (1)-(4) *are satisfied.*

Algorithm 6 allocates any unallocated edges from the allocation of Algorithm 5. If $X_0 = \emptyset$, then all remaining unallocated edges are allocated to X_0. If $X_0 \neq \emptyset$, we carefully select two non-envied neighbors of vertex 0 and the unallocated edges are allocated among those two vertices and 0 in such a way that no further envy is created.

Lemma 4.5. *If $\mathbf{X}$ is the allocation returned by Algorithm 5, then Algorithm 6 returns a complete EFX allocation.*

The running time of Subroutine A is polynomial on the number of different values of the social welfare, yet, this number can be exponential to the size of the input.

Lemma 4.6. *The construction of the EFX allocation by running Algorithms 5 and 6 needs pseudo-polynomial time.*

5 Conclusion

Our results build upon the existing literature on graph setting based on the definition of (Christodoulou et al. 2023), and push the state-of-the-art even further towards the general problem of EFX existence, which is equivalent to the multi-hypergraph setting without any restrictions. We introduce a different angle on the approach of the problem by designating from the start special vertices to allocate non-oriented edges. We remark that our approach is quite general and is applied to arbitrarily heterogeneous monotone valuations, and any restrictions come merely from the structure of the multi-hypergraph. Future directions include lifting the restrictions in the structure of the multi-hypergraph, or even improve complexity for restricted cases.

Acknowledgments

The research project is implemented in the framework of H.F.R.I call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union-NextGenerationEU (H.F.R.I. Project Number:15635). This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

Afshinmehr, M.; Danaei, A.; Kazemi, M.; Mehlhorn, K.; and Rathi, N. 2025. EFX Allocations and Orientations on Bipartite Multi-graphs: A Complete Picture. In Das, S.; Nowé, A.; and Vorobeychik, Y., eds., *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, 32–40. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Akrami, H.; Alon, N.; Chaudhury, B. R.; Garg, J.; Mehlhorn, K.; and Mehta, R. 2023. EFX: A Simpler Approach and an (Almost) Optimal Guarantee via Rainbow Cycle Number. In Leyton-Brown, K.; Hartline, J. D.; and Samuelson, L., eds., *Proceedings of the 24th ACM Conference on Economics and Computation, EC 2023, London, United Kingdom, July 9-12, 2023*, 61. ACM.

Amanatidis, G.; Birmpas, G.; Filos-Ratsikas, A.; Hollender, A.; and Voudouris, A. A. 2021. Maximum Nash welfare and other stories about EFX. *Theor. Comput. Sci.*, 863: 69–85.

Amanatidis, G.; Filos-Ratsikas, A.; and Sgouritsa, A. 2024. Pushing the Frontier on Approximate EFX Allocations. In *Proceedings of the 25th ACM Conference on Economics and Computation, EC 2024, New Haven, United States of America, July 8 - 11, 2024*.

Aziz, H.; and Mackenzie, S. 2016. A discrete and bounded envy-free cake cutting protocol for four agents. In Wichs, D.; and Mansour, Y., eds., *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. ACM.

Babaioff, M.; Ezra, T.; and Feige, U. 2021. Fair and Truthful Mechanisms for Dichotomous Valuations. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 5119–5126. AAAI Press.

Berge, C. 1973. *Graphs and Hypergraphs*. Amsterdam: North-Holland.

Berger, B.; Cohen, A.; Feldman, M.; and Fiat, A. 2022. Almost Full EFX Exists for Four Agents. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 4826–4833. AAAI Press.

Bhaskar, U.; and Pandit, Y. 2024. EFX Allocations on Some Multi-graph Classes. arXiv:2412.06513.

Budish, E. 2010. The combinatorial assignment problem: approximate competitive equilibrium from equal incomes. In Dror, M.; and Sosic, G., eds., *Proceedings of the Behavioral and Quantitative Game Theory - Conference on Future Directions, BQGT '10, Newport Beach, California, USA, May 14-16, 2010*, 74:1. ACM.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Economics and Comput.*, 7(3): 12:1–12:32.

Chaudhury, B. R.; Garg, J.; and Mehlhorn, K. 2024. EFX Exists for Three Agents. *J. ACM*, 71(1): 4:1–4:27.

Chaudhury, B. R.; Kavitha, T.; Mehlhorn, K.; and Sgouritsa, A. 2021. A Little Charity Guarantees Almost Envy-Freeness. *SIAM J. Comput.*, 50(4): 1336–1358.

Christodoulou, G.; Fiat, A.; Koutsoupias, E.; and Sgouritsa, A. 2023. Fair allocation in graphs. In Leyton-Brown, K.; Hartline, J. D.; and Samuelson, L., eds., *Proceedings of the 24th ACM Conference on Economics and Computation, EC 2023, London, United Kingdom, July 9-12, 2023*, 473–488. ACM.

Christodoulou, G.; and Mastrakoulis, S. 2025. Exact and approximate maximin share allocations in multi-graphs. *CoRR*, abs/2506.20317.

Deligkas, A.; Eiben, E.; Goldsmith, T.-L.; and Korchemna, V. 2024. EF1 and EFX Orientations. arXiv:2409.13616.

Feige, U. 2025. From multi-allocations to allocations, with subadditive valuations. arXiv:2506.21493.

Foley, D. K. 1966. *Resource allocation and the public sector*. Yale University.

Gamow, G.; and Stern, M. 1958. *Puzzle-math*. Viking Press. ISBN 9780670583355.

Hosseini, H.; Sikdar, S.; Vaish, R.; and Xia, L. 2021. Fair and Efficient Allocations under Lexicographic Preferences. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 5472–5480. AAAI Press.

Hsu, K. 2024. EFX Orientations of Multigraphs. arXiv:2410.12039.

Hv, V. P.; Ghosal, P.; Nimbhorkar, P.; and Varma, N. 2025. EFX Exists for Three Types of Agents. In *Proceedings of the 26th ACM Conference on Economics and Computation*, EC '25, 101–128. New York, NY, USA: Association for Computing Machinery. ISBN 9798400719431.

Kaviani, A.; Keshavarz, A.; Seddighin, M.; and Shahrezaei, A. 2025. Improved Approximate EFX Guarantees for Multigraphs. arXiv:2506.09288.

Kaviani, A.; Seddighin, M.; and Shahrezaei, A. 2024. Almost Envy-free Allocation of Indivisible Goods: A Tale of Two Valuations. In *Web and Internet Economics - 20th International Conference, WINE 2024, Edinburgh, United Kingdom, December 2-5*.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In Breese, J. S.; Feigenbaum, J.; and Seltzer, M. I., eds., *Proceedings 5th ACM Conference on Electronic Commerce (EC-2004), New York, NY, USA, May 17-20, 2004*, 125–131. ACM.

Plaut, B.; and Roughgarden, T. 2020. Almost Envy-Freeness with General Valuations. *SIAM J. Discret. Math.*, 34(2): 1039–1068.

Procaccia, A. D. 2020. An answer to fair division's most enigmatic question: technical perspective. *Commun. ACM*, 63(4): 118.

Sgouritsa, A.; and Sotiriou, M. M. 2025. On the Existence of EFX Allocations in Multigraphs. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, 2735–2737. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400714269.

Steihaus, H. 1948. The problem of fair division. *Econometrica*, 16: 101–104.

Stromquist, W. R. 1980. How to Cut a Cake Fairly. *American Mathematical Monthly*, 87: 640–644.

Varian, H. R. 1974. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1): 63–91.

Woodall, D. 1980. Dividing a cake fairly. *Journal of Mathematical Analysis and Applications*, 78(1): 233–247.

Zeng, J. A.; and Mehta, R. 2024. On the structure of envy-free orientations on graphs. *CoRR*, abs/2404.13527.

Zhou, Y.; Wei, T.; Li, M.; and Li, B. 2024. A Complete Landscape of EFX Allocations on Graphs: Goods, Chores and Mixed Manna. In Larson, K., ed., *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 3049–3056. Jeju, Korea: International Joint Conferences on Artificial Intelligence Organization. Main Track.