Autonomous Agents and Multi-Agent Systems (2026) 40:32
https://doi.org/10.1007/s10458-026-09754-8

EFX allocations and orientations on bipartite multi-graphs: a complete picture

Mahyar Afshinmehr[1] · Alireza Danaei[2] · Mehrafarin Kazemi[3] · Kurt Mehlhorn[4] · Nidhi Rathi[5]

Received: 15 October 2025 / Accepted: 4 May 2026

Abstract

We consider the fundamental problem of *fairly* allocating a set of indivisible items among agents having valuations that are represented by a *multi-graph* – here, agents appear as vertices and items as edges between them and each vertex (agent) only values the set of its incident edges (items). The goal is to find a fair, i.e., *envy-free up to any item* (EFX) allocation. This model has recently been introduced by [22] where they show that EFX allocations always exist on simple graphs for *monotone* valuations, i.e., where any two agents can share at most one edge (item). A natural question arises as to what happens when we go beyond simple graphs and study various classes of multi-graphs? We answer the above question affirmatively for the valuation class of *bipartite multi-graphs* and *multi-cycles*. The main contribution of this work is to establish the existence of EFX allocations on bipartite multi-graphs for *monotone* valuations and on multi-cycles for MMS-*feasible* valuations. We also present pseudo-polynomial time algorithms to compute EFX allocations for the above settings. Furthermore, we show that for bipartite multi-graphs with *cancelable* valuations, EFX allocations can be computed in polynomial time. We thus deepen the understanding of EFX allocations by expanding the spectrum of settings in which they are guaranteed to exist for an arbitrary number of agents. Next, we study EFX *orientations* (allocations where every item is assigned to one of its two endpoint agents) and provide a complete characterization of their existence on bipartite multi-graphs in terms of two key parameters—the number of edges shared between any two agents and the diameter of the graph. Finally, we prove that it is NP-complete to determine whether a given fair division instance on a bipartite multi-graph admits an EFX orientation, even with a constant number of agents.

Mahyar Afshinmehr: A part of the work was done when MA was an intern at MPI-INF, Germany.

Mehrafarin Kazemi: A part of the work was done when MK was an intern at MPI-INF, Germany.

Nidhi Rathi: A part of the work was done when NR was a Lise Meitner postdoctoral fellow at MPI-INF, Germany.

Extended author information available on the last page of the article

Published online: 02 June 2026 Springer

1 Introduction

The theory of *Fair Division* formalizes the classic problem of dividing a collection of resources to a set of participating players (referred to as *agents*) in a *fair* manner. This problem forms a key concern in the design of many social institutions and arises naturally in many real-world scenarios such as dividing business assets, assigning computational resources in a cloud computing environment, air traffic management, course assignments, divorce settlements, and so on [15, 26, 36, 38, 45]. The theory of fair division lies at the crossroads of economics, social science, mathematics, and computer science, with its formal exploration beginning in 1948 by Steinhaus, Banach, and Knaster. In recent years, this field has experienced a flourishing flow of research; see [7, 12, 13, 40] for excellent expositions.

In this work, we study the *discrete* setting of fair division where we wish to divide a set of *m* indivisible items to a set $\{1, 2, \dots, n\}$ of *n* agents with each item being allocated *wholly* to some agent. The standard notion of fairness in this line of work is that of *envy-freeness*, which entails a division (allocation) as *fair* if every agent values her bundle as much as any other bundle in the allocation [28]. Since envy-free allocations may not always exist for the case of indivisible items,[1] several variants of envy-freeness have been explored in the literature. Among all, *envy-freeness up to any item* (EFX) is considered to be the flag-bearer of fairness for the discrete setting (introduced by [16]). We say an allocation $X = (X_1, X_2, \dots, X_n)$ with bundle X_i being allocated to agent $i \in [n]$ is EFX if for every pair $i, j \in [n]$ of agents, agent *i* prefers her bundle X_i over $X_j \setminus \{g\}$ for *every* item $g \in X_j$. EFX is considered to be the "closest analogue of envy-freeness" for the discrete setting [17].

It remains a major open problem to understand the existence of EFX allocations [39]. Despite significant efforts, it is not known whether EFX allocations exist for four or more agents, even with additive valuations [20]. This naturally points towards the complexity of this problem and motivates the study of its various kinds of relaxations. That is, one may begin to understand the concept of EFX by finding approximate EFX allocations, by studying special valuation classes, or by relaxing the notion of EFX via charity or its epistemic form. We refer the readers to Section 1.2 for further discussion.

A question of interest is to understand for what settings or valuation classes, EFX allocations are guaranteed to exist. To this effect, in this work, we consider a recently introduced model by [22] where agent valuations are represented via a graph (or a multi-graph). Here, each item can have a positive value for at most two agents, and such an item is represented by an edge connecting those two agents. Equivalently, items are valued at zero by all agents who are not their endpoints. Thus, the vertices of the multi-graph correspond to agents, and the multi-edges between any pair of agents represent the items valued exclusively by them. For each agent $i \in [n]$, we assume a valuation function v_i such that an edge *g* has a positive marginal value ($v_i(S \cup g) - v_i(S) > 0$ for every subset *S*) if and only if *g* is incident to *i*. We call this a *fair division problem on multi-graphs* with the goal of finding complete EFX allocations.

This model captures situations in which the number of agents interested in any given item is limited—specifically, when an item is *relevant* or *valuable* to at most two agents. The graph-based formulation is inspired by geographic contexts where agents care only about

[1] Consider an instance with two agents valuing a single item positively. Here, the agent who does not receive the item will envy the other. In contrast, envy-free allocations always exist when the resource to be allocated is divisible (see [43, 44]).

nearby resources. For example, a trade corridor or a natural gas pipeline benefits only the bordering countries involved, while others gain nothing.

Christodoulou et al. [22] proved that EFX allocations exist for all simple graphs, where any pair of agents vertices share at most *one* edge item. This naturally raises the question of what happens in more general classes of multi-graphs, where a pair of agents can have multiple valuable items in common?

Christodoulou et al. [22] also introduced the notion of EFX *orientations* defined as EFX allocations with the extra condition that every item can be only allocated to one of its corresponding endpoints. They provided an example where EFX orientations do not exist in the simple graph setting. Christodoulou et al. [22] also proved that deciding whether a simple-graph instance admits an EFX orientation is NP-complete. A natural open question raised in this direction is that when do such orientations are guaranteed to exist for the multi-graph setting?

1.1 Our contribution and techniques

In this work, we answer the above question raised by [22] and study fair division instances where agent valuations are represented via *bipartite multi-graphs* and provide a complete picture of EFX allocations for such instances. Note that bipartite multi-graphs are a huge class consisting of multi-trees, multi-cycles with even length, and planar multi-graphs with faces of even length, to name a few.

We also study special types of (*non-wasteful*) allocations—EFX *orientations*—where every item is allocated to one of the agent endpoints. Note that EFX orientations are desirable since every item is allocated to an agent who values it, and hence, is *non-wasteful*. Our main results are as follows.

- The primary contribution of this work is to establish the existence of EFX allocations for fair division instances on *bipartite multi-graphs* with general *monotone* valuations (Theorem 4.11). Moreover, we can compute an EFX allocation in pseudo-polynomial time for monotone valuations and in polynomial time for *cancelable* valuations (Theorem 4.11). It is relevant to note that *cancelable* valuations are a strict superclass of additive valuations (see Section 2 for definitions).
  Hence, we answer the open problem listed in [22] and deepen our understanding of the existence of EFX allocations.
- While extending our techniques beyond bipartite multi-graphs, we also prove that EFX allocations exist and can be computed in pseudo-polynomial time on *multi-cycle graphs* with MMS-*feasible* valuations (Theorem 5.1). Note that MMS-*feasible* valuations are a strict superclass of cancelable valuations.
  We also show that for multi-cycles with at least four agents, EFX allocations always exist and can be computed in pseudo-polynomial time for monotone valuations and in polynomial time for cancelable valuations (Corollary 5.2).
- Next, we show that EFX *orientations* may not always exist for fair division instances on bipartite multi-graphs; in particular, they may not exist even in very simple settings of multi-trees. Nonetheless, we provide a complete characterization of the existence of EFX orientations for monotone valuations depending on two key parameters: q, the maximum number of edges shared between any two agents, and $d(G)$, diameter of the graph G (see Table 1).

Table 1 A complete picture for EFX orientations on bipartite multi-graphs with additive valuations based on q and $d(G)$, where q represents the maximum number of items between a pair of agent and $d(G)$ represents the diameter of the input graph instance

Parameters	EFX Orientation
acyclic, $q = 2$, $d(G) \leq 4$	Exists (Theorem 3.3)
acyclic, $q = 2$, $d(G) > 4$	May not exist (Theorem 3.5)
acyclic, $q > 2$, $d(G) \leq 2$	Exists (Proposition 4.3)
acyclic, $q > 2$, $d(G) > 2$	May not exist (Theorem 3.4)
cyclic, $q \geq 2$, $d(G) \geq 2$	May not exist (Theorem 3.2)

The fact that EFX orientations do not always exist can be seen as a proof to show that inefficiency is inherent and hence approximations are necessary. We show that, for additive valuations, there exist orientations on bipartite multi-graphs where at least $\lceil \frac{n}{2} \rceil$ agents are EFX and the remaining agents are 1/2-EFX (Theorem 4.12).

- We also show that for monotone valuations, we can compute EFX orientations in polynomial time when the diameter of the acyclic bipartite multi-graph is at most four and any two adjacent vertices share at most two edge items (Theorem 3.3).
- It is NP-complete to decide whether a given fair division instance on bipartite multi-graphs admits an EFX orientation, even with a constant number of agents (Theorem 3.6).

Two other papers independently and in parallel showed the existence of EFX allocations involving multi-graphs. Sgouritsa and Sotiriou [41] independently show that the EFX allocations always exist in bipartite multi-graphs for monotone valuations. This coincides with our first result, but we further prove polynomial-time computability of EFX allocations (in the above setting) for cancelable valuations and in pseudo-polynomial time for monotone valuations. Moreover, Sgouritsa and Sotiriou [41] proves the existence of EFX allocations for multi-graphs when each agent has at most $\lceil n/4 \rceil - 1$ neighbors, where n is the total number of agents, or when the shortest cycle with non-parallel edges has length at least 6.

Bhaskar and Pandit [11] independently showed the existence of EFX allocations in bipartite multi-graphs for cancelable valuations and in multi-trees for general monotone valuations. Our first result generalizes these two results. Moreover, they showed that multi-graphs with chromatic number t admit EFX allocations when the shortest cycle using non-parallel edges is of length at least $2t - 1$; this result holds when agents have cancelable valuations. Note that the multi-graphs with chromatic number 2 are just the bipartite multi-graphs.

Technical overview We will give a description of the main techniques developed in this work to prove the existence of EFX allocations on bipartite multi-graphs with monotone valuations. For simplicity, we will assume additive valuations here. For a given bipartite multi-graph $G = (S \sqcup T, E)$, let us consider two adjacent vertices $i \in S$ and $j \in T$. The starting point in our technique is inspired by the *cut-and-choose* protocol that is used to prove the existence of EFX allocations for two agents.[2] Based on this protocol, we will define one specific *configuration* for the set of items, $E(i, j)$, between i and j. In this configuration, $j \in T$ will cut the set $E(i, j)$ into two bundles C_1 and C_2 such that she is EFX-happy

[2] For two agents, the first agent divides the set of items into two EFX-feasible bundles (for her), and the second agent chooses her favorite bundle of the two, while the other bundle goes to the first agent.

with both bundles. We call this the *j-cut configuration* between agents *i* and *j*. Note that such a configuration can be computed in polynomial time for additive valuations.

The idea is to find a partial EFX orientation *X* that satisfies a set of six useful properties (listed in Section 4.2). These properties pave a simple way for us to extend *X* to a complete allocation while maintaining EFX guarantees. One of them ensures that for any $i \in S, j \in T$, either (i) no item from *E*(*i*, *j*) is allocated, (ii) exactly one of C_1 or C_2 is allocated to either *i* or *j*, or (iii) both C_1 and C_2 are allocated to *i* and *j*, such that one receives C_1 and other C_2 in *X*.

To create such a partial orientation, we start with a greedy algorithm that allocates a set of items to each agent such that every agent in *T* is non-envied. This initial step ensures that the set X_i of items allocated to any agent $i \in [n]$ is such that $X_i \subseteq E(i, j)$ for some $j \in [n]$. Moreover, this partial orientation is EFX wherein any vertex that has envy is certainly non-envied. We then try to orient unallocated items incident to a non-envied vertex to her while maintaining partial EFX until the only unallocated edges are between a non-envied and an envied vertex.

Once we find a partial EFX orientation with certain useful properties, we can go two ways (of our choice) to have a complete allocation. We can either compute, in polynomial time, (i) an orientation where at least *n*/2 agents are EFX, and the remaining agents are 1/2-EFX (Theorem 4.12), or (ii) an exact EFX allocation (Theorem 4.11). For the latter, we know that since EFX orientations do not necessarily exist for bipartite multi-graphs, we have to allocate the remaining edges to a vertex other than their endpoints, which will create a wasteful (albeit an EFX) allocation.

Before finding a complete allocation, we have ensured, by one of our properties, that both non-envied and envied vertices are satisfied enough with what is allocated to them that they will not envy if we give all the unallocated edges adjacent to them to a specific third vertex. We, therefore, safely allocate the remaining items from the set *E*(*i*, *j*) to a specific agent $k \neq i, j$, and finally compute an EFX allocation.

1.2 Further related work

For the notion of EFX, Plaut and Roughgarden [37] proved its existence for two agents with monotone valuations. For three agents, a series of works proved the existence of EFX allocations when agents have additive valuations [20], *nice-cancelable* valuations [10], and finally when two agents have monotone valuations and one has an MMS-*feasible* valuation [4]. EFX allocations exist when agents have identical [37], binary [31], or bi-valued [6] valuations. The study of several approximations [5, 19, 21, 27] and relaxations [3, 6, 9, 10, 17, 18, 32, 34] of EFX have become an important line of research in discrete fair division.

Another relaxation of envy-freeness proposed in discrete fair division literature is that of *envy-freeness up to some item* (EF1), introduced by [14]. It requires that each agent prefers her own bundle to the bundle of any other agent after removing some item from the latter. EF1 allocations always exist and can be computed efficiently [33]. *Epistemic* EFX is another relaxation of EFX that was recently introduced by [18], where they showed its existence and polynomial-time tractability for additive valuations. A recent work [2] then established the existence of epistemic EFX allocations for monotone valuations.

Following the work of [22], recent works have started to focus on EFX and EF1 orientations and allocations on graph settings. Zhou et al. [48] studies the mixed manna setting with both goods and chores and proves that determining the existence of EFX orientations on simple graphs for agents with additive valuations is NP-complete and provides certain special cases like trees, stars, and paths where it is tractable. Zeng and Mehta [46] relates the existence of EFX orientations and the chromatic number of the graph. Recently, Deligkas et al. [24] showed that EF1 orientations always exist for monotone valuations and can be computed in pseudo-polynomial time. Moreover, Deligkas et al. [24] proved that computing EFX orientations remains intractable even under strong structural restrictions such as bounded vertex cover or a constant number of vertices; nevertheless, a nearly optimal fixed-parameter tractable algorithm is known by [24].

Proportionality [25, 42] and *maximin fair share* [14], aka MMS, are two other important fairness notions; we refer the readers to an excellent recent survey by [7] (and references within) for further discussion.

Research on fair division has also extended to the setting of *indivisible chores*, where agents incur *costs* rather than receive utilities. In this setting, substantial progress has been made toward understanding approximate EFX fairness. The early work of [47] achieved an $O(n^2)$-approximation for additive cost functions, and subsequent improvements by [30] tightened this to a constant 4-EFX guarantee for any number of agents. Parallel developments by [23] and [1] established the existence of 2-EFX allocations for three agents, and [29] later generalized this result to arbitrary numbers of agents. Despite these advances, exact EFX allocations remain known only in highly restricted settings. In fact, Christoforidis and Santorinaios [23] demonstrated that for monotone cost functions, some instances admit no EFX allocation at all. Meanwhile, a long-standing open question on the compatibility of fairness and efficiency—whether an allocation can be both EF1 and PO—was recently answered in the affirmative by [35].

Roadmap We begin by introducing notation and definitions in Section 2. In Section 3, we develop an exhaustive list of scenarios where EFX orientations exist depending on two parameters related to multi-graphs for monotone valuations. In Section 4, we develop a pseudo-polynomial-time algorithm to compute EFX allocations for bipartite multi-graphs with monotone valuations. Finally, in Section 5, we extend our results and develop a pseudo-polynomial-time algorithm to compute EFX allocations for multi-cycles with MMS -feasible valuations.

2 Notation and definitions

For any positive integer k, we use $[k]$ to denote the set $\{1, 2, \ldots, k\}$. We consider a set $[m]$ of m goods (items) that needs to be allocated among a set $[n] = \{1, 2, \ldots, n\}$ of n agents in a fair manner. For ease of notation, we will use g instead of $\{g\}$ for an item $g \in [m]$.

Definition 2.1 (Allocations). A partial allocation $X = (X_1, X_2, \ldots, X_n)$ is an ordered tuple of n disjoint subsets of $[m]$, i.e., for every $i, j \in [n]$ we have $X_i \cap X_j = \emptyset$. Here, X_i denotes the bundle allocated to agent $i \in [n]$ in X. We say an allocation X is complete if $\bigcup_{i \in [n]} X_i = [m]$.

Valuation functions and fairness notions Each agent $i \in [n]$ specifies her preferences using a valuation function $v_i : 2^{[m]} \to \mathbb{R}^+$, that assigns a non-negative value to every subset of items. We denote a fair division instance as $\mathcal{I} = \left\langle [n], [m], \{v_i\}_{i \in [n]} \right\rangle$.

This work focuses on monotone, additive, cancelable, and MMS-feasible valuations that can be represented via *multi-graphs*, defined below.

Definition 2.2 (Monotone, Additive, Cancelable, and MMS-feasible Valuations). We say a valuation function $v : 2^{[m]} \to \mathbb{R}^+$ is monotone if for every $S \subseteq S' \subseteq [m]$, we have $v(S) \leq v(S')$. We say *v* is *additive* if for every subset $S \subseteq [m]$ of items, we have $v(S) = \sum_{g \in S} v(g)$.

Next, we say *v* is cancelable if for any $S, T \subseteq [m]$ and any $g \in [m] \setminus (S \cup T)$, we have

$$v(S \cup \{g\}) > v(T \cup \{g\}) \implies v(S) > v(T).$$

That is, removing the same good from two bundles would not change the relative preference between the two. Note that the class of cancelable valuations is a strict superclass of additive valuations.

Finally, we say *v* is MMS-feasible if for any $S \subseteq [m]$ and any partitions $A = (A_1, A_2)$ and $B = (B_1, B_2)$ of *S*, we have $\max(v(B_1), v(B_2)) \geq \min(v(A_1), v(A_2))$. Intuitively, MMS-feasibility ensures that no partition of a set is "strictly better" in a balanced sense than another partition. In every two-way split, the larger bundle in one partition is at least as valuable as the smaller bundle in any other partition, preventing extreme imbalances across partitions.

Note that the class of MMS-feasible valuations is a strict superclass of cancelable valuations.

Recently, Christodoulou et al. [22] proposed a valuation class that can be represented by graphs. Here, vertices correspond to agents and edges correspond to items that are valued positively only by the two agents at their endpoints. Christodoulou et al. studied simple graphs wherein there is at most one edge between every pair of adjacent vertices. In this work, we focus on a natural extension of these instances to *multi-graphs* where we allow multiple edges between agents.

Definition 2.3 (Multi-graph Instances). A fair division instance $\mathcal{I} = \left\langle [n], [m], \{v_i\}_{i \in [n]} \right\rangle$ on a multi-graph[3] is represented via a multi-graph $G = (V, E)$ where agents form the set $V = [n]$ of vertices and items form *m* edges in *E* with the following structure: for every agent $i \in [n]$ and every subset $S \subseteq [m]$, $v_i(S) = v_i(S \cap E(i))$, where *E*(*i*) denotes the set of items incident to agent *i*, i.e., an item $g \in [m]$ has a positive marginal value for agent *i* if and only if *g* is incident to *i*.

We denote the set of edges between agents *i* and *j* by *E*(*i*, *j*). Note that $E(i, j) = E(j, i)$. Also, we use the words 'agent' and 'vertex' interchangeably, similarly for 'item' and 'edge'.

[3] A multi-graph can have multiple edges between two vertices.

Definition 2.4 (Symmetric Instances) We say a multi-graph instance is symmetric if the valuations are additive and for any item $g \in E(i, j)$ with $i, j \in [n]$ is identically valued by both *i* and *j*, i.e., $v_i(g) = v_j(g)$.

Let us now define the concept of non-wasteful allocations known as *orientations*.

Definition 2.5 (Partial Orientation). A partial orientation is a partial allocation where an item *g* (if allocated) is given to an agent *i* such that *g* is incident to *i* in the given multi-graph. This can be represented by directing the (allocated) edges in the graph towards the vertex receiving the edge.

We use the popular notion of *envy-freeness up to any good* (EFX) as a standard fairness notion in our work. Let us first define the concept of *strong envy* and some useful constructs related to envy.

Definition 2.6 (Envy and Strong Envy). Given an allocation $X = (X_1, X_2, \ldots, X_n)$, we say *i* envies *j* if $v_i(X_j) > v_i(X_i)$, and we say *i strongly envies j* if there exists an item $g \in X_j$ such that $v_i(X_i) < v_i(X_j \setminus g)$.

Definition 2.7 (Envy-Freeness Up to Any Good (EFX)). We say an allocation is EFX if there is no strong envy between any pair of agents. Moreover, we say an allocation *X* is α -EFX for an $\alpha \in (0, 1]$ if $v_i(X_i) \geq \alpha \cdot v_i(X_j \setminus g)$ for every $i, j \in [n]$ and $g \in X_j$.

Definition 2.8 (EFX-Feasibility). Given a partition $X = (X_1, X_2, \ldots, X_n)$ of items into *n* bundles, we say bundle X_k is EFX-feasible for agent *i* if we have $v_i(X_k) \geq \max_{j \in [n]} \max_{g \in X_j} v_i(X_j \setminus g)$.

We say a bundle containing one item as a "singleton". Note that no agent strongly envies an agent owning a singleton. Also, since, in an orientation on a given multi-graph *G*, a vertex $i \in [n]$ receives edges that are incident to her, it implies that the set of *i*'s neighbors in *G* are the only agents that can possibly strongly envy her. This leads to the following observation, which we will use frequently in future sections.

Observation 2.9 *A partial orientation is* EFX *on a multi-graph if and only if no agent strongly envies her neighbor.*

Next, we demonstrate a property of EFX orientations on multi-graphs via the following lemma.

Lemma 2.10 *For a multi-graph instance, consider a partial* EFX *orientation X. If a vertex i is envied by one of its neighbors j, then it must hold that* $X_i \subseteq E(i, j)$. *In particular, in any* EFX *orientation, each vertex can be envied by at most one neighbor.*

Proof For contradiction, let us assume that there exists an agent *i*, who is envied by her neighbor agent *j* such that there exists an $e \in X_i$, where $e \in E(i, k)$ for some $k \neq j$. Note that $v_j(X_j) < v_j(X_i) = v_j(X_i \setminus e)$, contradicting the fact that *X* is EFX. □

2.1 Graph theory definitions

Our work characterizes the existence of EFX orientations based on the parameter q (the maximum number of edges between any of agents in G) and the diameter of the multi-graph. For every multi-graph instance, We begin by defining some useful notions related to a multi-graph.

Definition 2.11 (Skeleton of a Multi-graph). For a multi-graph $G = (V, E)$, we define its skeleton as a graph $G' = (V, E')$ where G' has the same set of vertices, and there is a single edge between two vertices if they share at least one edge in G, i.e., i is connected to j in G' if $E(i, j) \neq \emptyset$ in G.

Definition 2.12 ($d(G)$ and q of a Multi-graph G). We define $d(G)$ as the diameter of G, which is the length of the longest shortest path in the skeleton of G. And, we denote $q = \max_{i,j\in[n]} |E(i, j)|$ to be the maximum number of edges between any pair of agents in G.

Definition 2.13 (Center of a Multi-graph). For a multi-graph G, center $c \in V$ is a member of the set $\text{argmin}_{x\in V} \max_{v\in V} d(x, v)$, where $d(x, v)$ is the distance between x and v in G. In this paper, we choose one arbitrarily if we have multiple centers.

In this work, we focus on bipartite multi-graphs, that we define next.

Definition 2.14 (Bipartite Multi-graph). A bipartite multi-graph $G = (V, E)$ has a skeleton that is a bipartite graph. We denote $V = S \sqcup T$ where S and T are two partitions with only edges occurring between vertices in S and vertices in T

Definition 2.15 (Multi-star, Multi-P_n, Multi-cycle, and Multi-tree). A multi-H has a skeleton that is an H graph, where H can be a star, a path P_n of length $n - 1$, a cycle, or a tree.

Note that bipartite multi-graphs are an important class consisting of multi-trees, multi-cycles of even length, and planar multi-graphs with faces of even length, to name a few.

3 EFX orientations on bipartite multi-graphs

Christodoulou et al. shows that EFX orientations may not always exist, even on simple graphs. Therefore, it is not surprising when we show the same on bipartite multi-graphs. In particular, we examine multi-cycles (a special kind of bipartite multi-graphs) and show that even for $q = 2$, EFX orientations may not always exist for four agents; see Fig. 1. Nonetheless, we identify the correct parameters to characterize the scenarios where EFX orientations are guaranteed to exist for monotone valuations; see Table 1.

To do so, we proceed step by step, carefully considering all possible cases. Initially, we focus on bipartite multi-graphs with diameters, $d(G)$, of small numbers. For $d(G) = 1$, bipartite multi-graphs become multi-P_2, for which an EFX orientation can easily be achieved using the cut-and-choose protocol. As a warm-up, we show the existence of EFX orientations for multi-trees with $d(G) = 2$, i.e., for multi-stars. Here, we do so for $q = 2$, but the same approach can be generalized to any q, which we discuss in the next section (in Proposition 4.3).

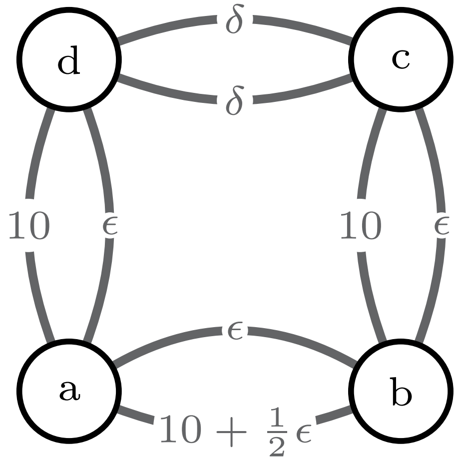

Fig. 1 Counter-example for Theorem 3.2

In this section, we present counter-examples for various cases, via figures, where we have symmetric instances, and the number on each edge depicts the value of that edge for both endpoints.

Proposition 3.1 EFX *orientations exist on multi-stars for* $q = 2$.

Proof Let r be the center of the given multi-star instance. Every leaf i chooses the edge she prefers the most from the two edges in $E(i, r)$, and we allocate the remaining edges to agent r, hence creating an orientation. By Observation 2.9, note that to prove that the allocation is EFX, we only need to check if the EFX condition is satisfied between agent r and her neighbors. Agent r does not strongly envy any leaf because every leaf has a singleton edge in her bundle. Moreover, every leaf receives her preferred item from $E(i, r)$, so she will not envy agent r. This completes our proof. □

However, an EFX orientation might not always exist on bipartite multi-graphs with $d(G) \geq 2$. We prove it by providing an example in the following theorem; see Fig. 1.

Theorem 3.2 EFX *orientations on cyclic bipartite multi-graphs with* $d(G) \geq 2$ *and any* $q \geq 2$ *may not exist (even on symmetric instances).*

Proof We use Fig. 1 to represent our counter-example (on multi-cycles with four agents) where the numbers (with $1 \gg \epsilon \gg \delta \approx 0$) on each edge show the value of that edge for both endpoints. We will try to construct an EFX orientation and show that it is not possible.

Without loss of generality, we assume the edge with value $10 + \frac{1}{2}\epsilon$ between agents a and b is allocated to agent a. First, let us suppose that the other edge in $E(a, b)$ is also allocated to a. For agent b to not strongly envy a, the bundle $E(b, c)$ must be fully allocated to b. This leaves us no way to allocate items to c so that she does not strongly envy b, reaching a contradiction. Therefore, the other edge in $E(a, b)$ must be allocated to b.

Now, consider the edge with value 10 in $E(b, c)$. If we allocate it to b, then c will strongly envy her. Therefore, this edge must be allocated to c. Observe that no other edge can be allocated to c; otherwise, b will strongly envy c. The same argument goes for a, and hence, it can only have one item of value $10 + \frac{1}{2}\epsilon$. Therefore, d must receive all her incident edges, making a strongly envy d. Therefore, no EFX-orientation exists for this instance.

It is easy to observe that the above counter-example can be extended by adding more edges with value δ between c and d or adding more vertices to the graph with edges of value $\delta' \approx 0$ to achieve a counter-example for bipartite multi-cyclic graphs with $q \geq 2$, and $d(G) \geq 2$. □

Since EFX orientations do not exist even on bipartite multi-cyclic graphs with small diameters, we examine the existence of EFX orientation on multi-trees as an important subset of bipartite multi-graphs. We show EFX orientations exist on multi-trees with $d(G) \leq 4$ and $q = 2$.

Theorem 3.3 EFX *orientations exist for monotone valuations and can be computed in polynomial time on multi-trees with* $d(G) \leq 4$ *and* $q = 2$.

Proof Let us denote the center of G as c and let it be rooted at c. We proceed via strong induction on the number of agents at depth 2 in G. We begin by making the induction statement stronger and prove that EFX orientations exist on multi-trees with $d(G) \leq 4$ and $q = 2$ while satisfying the following two properties:

1. For any envied agent i at depth one, the set $E(c, i)$ is completely allocated to either i or c.
2. For every envied agent i at depth one, i does not envy c.

We consider multi-stars for the base case of the induction where there are zero agents at depth 2. Let c pick her most valuable item. Then, we allocate all the remaining items incident to c to their other endpoint. In this orientation, properties (1) and (2) are satisfied. Also, the orientation is EFX because c owns a single item, so no one strongly envies her, and c does not strongly envy anyone because she receives her most preferred item.

Now, we move to the induction step. For every vertex t at depth one in G, let f_t be her most valued item in the set $\bigcup_{j \text{ is a child of } t \text{ in } G} E(t, j)$. For the induction step, consider an arbitrary vertex i at depth one and remove all her children from G. According to the induction hypothesis, an EFX orientation X exists for the remaining multi-tree that satisfies properties (1) and (2). Now, we add the removed vertices back and show how we extend X to an EFX orientation for G (by allocating the items in $\bigcup_{j \text{ is a child of } i \text{ in } G} E(i, j)$). We distinguish between three possible cases:

- i **is non-envied in** X: Here, we let every child j of i choose her preferred item in $E(i, j)$ and allocate the other edge to agent i. Agent i does not strongly envy her children because they own singletons. This gives an EFX orientation for G. Note that agent i remains non-envied, and using the induction hypothesis, we satisfy properties (1) and (2) here as well.
- i **is envied in** X **and** $v_i(E(c, i)) \geq v_i(f_i)$: Here, i is envied by c and by property (1), bundle $E(c, i)$ is allocated to i in X. For every child j of i, allocate the bundle $E(i, j)$ to agent j. Agent i does not strongly envy her children because she receives $E(c, i)$, which values at least as much as f_i. This gives an EFX orientation for G. Agent i is envied,

but we didn't reorient the edges in X; therefore, properties (1) and (2) are still satisfied using the induction hypothesis.

- *i* **is envied in** X **and** $v_i(E(c,i)) < v_i(f_i)$: Here, i is envied by c and by property (1), bundle $E(c, i)$ is allocated to i in X. Assume $f_i \in E(i, j_0)$, where j_0 is a child of i in G. Allocate f_i to i and give the other item in $E(i, j_0)$ to j_0. For every other child $j \neq j_0$ of i, allocate the bundle $E(i,j)$ to j. Reorient the edges in $E(c, i)$ and allocate all of them to c. Now, if agent c was non-envied in X, our resulting orientation will be EFX too. Also, one can observe that both properties are satisfied as well. Thus, we only consider the case where c was envied in X. By property (2), she can only be envied by a non-envied vertex at depth one. Let h be the vertex that was envying c in X. Reverse the orientation of edges between c and h. Now, i does not envy c. Also, c does not envy h because she received a more valuable bundle than her previous one. Therefore, h is still non-envied. One can observe that the resulting orientation is again EFX, and both properties are satisfied.

Overall, we can obtain an EFX orientation in G. Note that since our inductive step can be performed in polynomial time, we can compute an EFX orientation for G in polynomial time as well. □

Unfortunately, EFX orientations may not exist on multi-trees with a greater diameter or higher q, as shown in the following theorems.

Theorem 3.4 *For multi-trees with $d(G) \geq 3$,* EFX *orientations may not exist for $q \geq 3$ (even on symmetric instances).*

Proof We divide the counter-examples into two cases and show the non-existence of EFX orientations even for four agents with different q values (they are, in fact, multi-P_4 instances). We use Fig. 2 to represent our counter-examples.

Case 1. $q = 3$: Consider the instance in Fig. 2(a) and assume, for contradiction, that it admits an EFX orientation. Without loss of generality, let us suppose that agent b receives the edge (b, c). Agent a cannot receive the set $E(a, b)$ completely; otherwise, b will strongly envy a. Therefore, agent b receives at least one edge other than (b, c). Let us call this edge as e. The most valuable bundle that c can receive such that d does not strongly envy her is valued $2 + \epsilon$ to her. Therefore, we have,

$$v_c(X_b \backslash e) = v_c(E(b,c)) = 2 + \frac{3}{2}\epsilon > 2 + \epsilon \geq v_c(X_c),$$

which is a contradiction.

Case 2. $q \geq 4$: Consider the instance in Fig. 2(b) where there are q-many edges with value 1 for both endpoints between agents a and b and agents c and d. We assume, for contradiction, that this instance admits an EFX orientation. Again, without loss of generality, let agent b receive the edge (b, c). Agent a cannot receive the set $E(a, b)$ completely; otherwise, b will strongly envy a. Therefore, agent b receives at least one edge other than (b, c). Let us call this edge as e. The most valuable bundle that c can receive such that d does not strongly envy her is valued $\lceil \frac{q}{2} \rceil$ to her. Therefore, we have,

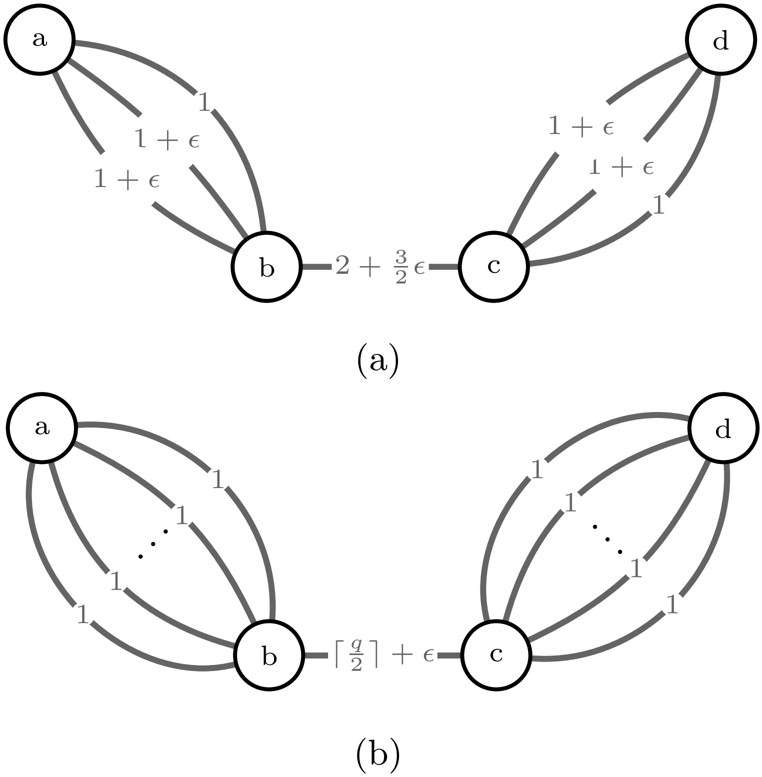

Fig. 2 Counter-example for Theorem 3.4

$$v_c(X_b \backslash e) = v_c(E(b,c)) = \lceil \frac{q}{2} \rceil + \epsilon > \lceil \frac{q}{2} \rceil \geq v_c(X_c),$$

which is a contradiction.

One can extend these counter-examples by adding nodes to the graph with edges with value $\delta \approx 0$ to achieve a counter-example for any $q \geq 3$ and $d(G) \geq 3$. □

Theorem 3.5 *For multi-trees with $d(G) \geq 5$,* EFX *orientations may not exist even for $q = 2$ (even on symmetric instances).*

Proof We present a multi-P_6 instance with $q = 2$ that does not admit any EFX orientation. We use Figs. 3 and 4 to represent our counter-example.

We first construct a special multi-P_3 instance where a specific node in any of its EFX orientations is envied. Consider the multi-P_3 instance in Fig. 3(a). One can easily show that it admits exactly two EFX orientations, and agent 3 is envied in both; see Fig. 3(b) and (c),

Now we build the multi-P_6 instance in Fig. 4 that is made by two copies of the above-mentioned special multi-P_3 instance. We connect them with an edge of value $\delta \ll \epsilon$ using the two vertices (vertex 3 and vertex 4) that are always envied in EFX orientations of the multi-P_3 parts. Without loss of generality, let agent 3 receive the edge (3, 4) with value δ. Now,

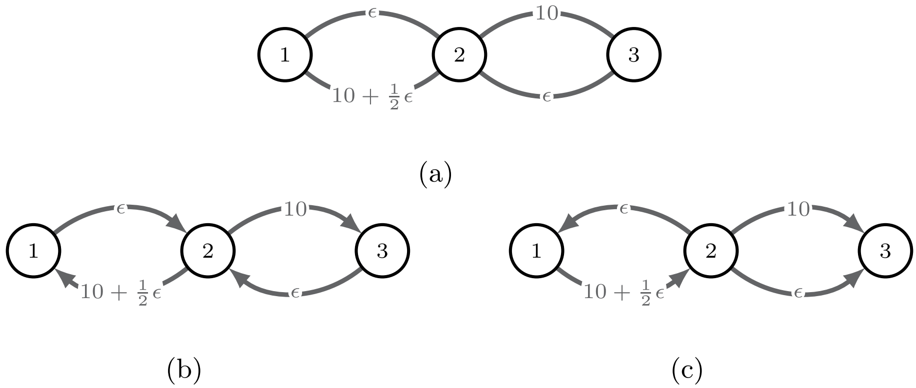

Fig. 3 We depict a multi-P_3 instance whose every EFX orientation leaves agent 3 envied. We use this instance as a building block to give a multi-P_6 instance that does not admit any EFX orientation. Orientations (**b**) and (**c**) are the only EFX orientations in this instance

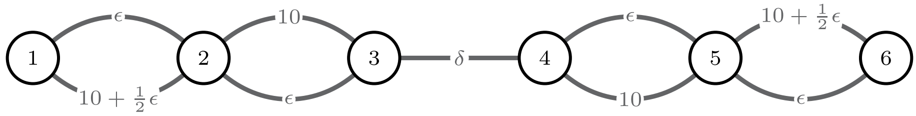

Fig. 4 A multi-P_6 instance that does not admit any EFX orientation

since in any EFX orientation of the special multi-P_3 instance, agent 3 was always envied, the addition of edge with value δ to agent 3's bundle will therefore create strong envy against her.

One can extend the above counter-example by adding nodes to the graph with edges having value $\delta' \approx 0$ to achieve a counter-example for any $d(G) \geq 5$. $\square$

3.1 Hardness of deciding the existence of EFX orientations

In this section, we consider the computational problem of deciding whether a given instance on a bipartite multi-graph (even a multi-tree with a constant number of agents) admits an EFX orientation. We reduce the NP-complete *Partition Problem* to our problem.

Partition problem Consider a multi-set[4] $P = \{p_1, p_2, \ldots, p_k\}$ of k non-negative integers. The problem is to decide whether P can be partitioned into two multi-sets P_1 and P_2 such that $\sum_{p \in P_1} p = \sum_{p \in P_2} p$.

We prove a stronger claim and prove hardness for multi-tree instances.

Theorem 3.6 *The problem of deciding whether a fair division instance on a multi-tree (with additive valuations) admits an* EFX *orientation is* NP*-complete. It holds true even for symmetric instances with a constant number of agents.*

[4] A multi-set allows multiple instances for each of its elements.

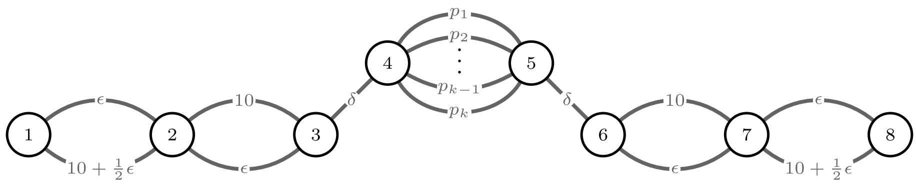

Fig. 5 The construction used in proof of Theorem 3.6. Here, $\delta \ll \epsilon \ll 1$

Proof Given an orientation, it is easy to verify if it is EFX. Hence, the problem belongs in NP.

Let us now consider an instance $P = \{p_1, p_2, \dots, p_k\}$ of the Partition problem. We will construct a fair division instance on a multi-tree with eight vertices, as depicted in Fig. 5. Note that we have used the multi-P_3 instance that appears in the proof of Theorem 3.5 here as well. Following the similar lines, we can argue that in any EFX orientation of this instance, agent 2 envies agent 3, and agent 7 envies agent 6. Thus, the two edges (3, 4) and (5, 6) are allocated to agents 4 and 5, respectively.

Next, we claim that this instance admits an EFX orientation if and only if we can partition the items in E(4, 5) such that agents 4 and 5 do not envy each other. Note that if agents 4 and 5 do not envy each other, then we can construct a complete EFX orientation as follows: take any EFX orientation for the multi-P_3 subgraphs containing only agents 1, 2, 3 and 4, 5, 6. Allocated the edge (3, 4) to agent 4 and (5, 6) to agent 5. Then, use the partition of E(4, 5) such that agents 4, 5 do not envy each other. Now, we prove the other direction via contradiction. Assume that there exists a complete EFX orientation such that at least one of the agents 4 or 5 envies the other. Without loss of generality, assume agent 5 envies agent 4. Note that since we have $X_4 \cap E(4,5) \neq \emptyset$, the allocation cannot be an EFX orientation by Lemma 2.10.

Now, let $P_1 \sqcup P_2 = P$ be a partition of items in E(4, 5). Note that if $\sum_{p \in P_1} p \neq \sum_{p \in P_2} p$, we can choose a $\delta > 0$ such that $\delta < |\sum_{p \in P_1} p - \sum_{p \in P_2} p|$. This will ensure that allocating P_1 and P_2 to agents 4 and 5 will cause envy only in one direction. Therefore, E(4, 5) can be partitioned in a way that agents 4 and 5 do not envy each other if and only if there exists a partition of P into two multi-sets P_1 and P_2 such that $\sum_{p \in P_1} p = \sum_{p \in P_2} p$. This completes the proof. □

Theorem 3.6 immediately implies the following, which is independently proved by [24]:

Corollary 3.7 *Deciding whether an EFX orientation exists on a multi-graph with additive valuations is* NP*-complete, even for symmetric instances with a constant number of agents.*

4 Existence of EFX allocations on bipartite multi-graphs for monotone valuations

In this section, we prove our main result showing that any fair division instance on a bipartite multi-graph with *monotone* valuations always admits an EFX allocation. Furthermore, such allocations can be computed in pseudo-polynomial time for *monotone* valuations and in polynomial time for *cancelable* valuations, a strict superclass of additive valuations (Theorem 4.11). Recall that (as discussed in Section 3), EFX orientations may not exist for these instances, and hence, we know that an EFX allocation would allocate some items to an agent who may not value them.

We will denote a fair division instance on a bipartite multi-graph by $G = (S \sqcup T, E)$, where S and T represent the two bi-partition parts of its skeleton. We begin by discussing the main idea of our technique (in Section 4.1) and then define some concepts and properties (in Section 4.2) useful for our proof (in Sections 4.3, 4.4, 4.5, 4.6, and 4.7). For the ease of understanding, we have further presented an example of executing our algorithm step by step on a bipartite multi-graph with additive valuations (see Appendix A).

4.1 Main idea

We introduce the concept of a *configuration* to decide how to allocate the edges between any two adjacent vertices (in G) to their endpoints. It has a flavor that is similar to the cut-and-choose protocol (used for finding EFX allocations between two agents). In our proof, we will use this configuration to partially orient the edges between two agents. We define it as follows:

Definition 4.1 ***T-cut Configurations:*** For any pair of agents $i \in S$ and $j \in T$, we let agent j to partition the set $E(i, j)$ into two bundles C_1 and C_2 such that both are EFX-feasible for j (for the items $E(i, j)$). We call the partition (C_1, C_2) as the *j-cut* configuration between agents i and j.

We now briefly discuss the running time of computing such partitions in the following remark, which will later be used to derive the overall running time of our algorithm.

Remark 4.2 It is well known that the algorithm referred to as the PR algorithm (using the terminology as [4]), first introduced by [37], can compute our cut configurations in pseudo-polynomial time for monotone valuations (see the proof of Theorem 3.49 in [8]). Moreover, [8] proved that with minor modifications, this algorithm can compute our desired cut configurations in polynomial time for cancelable valuations (see Lemma 4.7 in [8]).

As a warm-up, we will use *T-cut* configurations to prove the existence of EFX orientations for multi-stars with any q. Previously, (in Proposition 3.1), we proved the existence of EFX orientations for multi-stars with $q = 2$.

Proposition 4.3 EFX *orientations exist and can be computed in pseudo-polynomial time for multi-stars with any q. Moreover, we can compute such allocations in polynomial time for cancelable valuations.*

Proof Assume c is the center of the given multi-star, and assume that $T = \{c\}$ in G. For every vertex i incident to c, consider the *c-cut* configuration (C_1, C_2) between them. Then, we allocate the bundle which i prefers between C_1 and C_2 to her and allocate the other bundle to c. Therefore, for every such vertex i, she will not envy agent c. Moreover, since we are using the *c-cut* configurations, agent c will not strongly envy any of her neighbors. Using Remark 4.2, it is clear that the final orientation can be computed in pseudo-polynomial time for monotone valuations and in polynomial time for cancelable valuations. □

Table 2 Definition of $A_{i,j}(X)$ and $A_{j,i}(X)$

	$A_{i,j}(X)$	$A_{j,i}(X)$
$E(i,j) \cap X_k = \emptyset$ for all $k \in [n]$	$\arg\max_{C_1,C_2} \{v_i(C_1), v_i(C_2)\}$	$\arg\max_{C_1,C_2} \{v_j(C_1), v_j(C_2)\}$
$X_i \cap E(i,j) = \emptyset, X_j \cap E(i,j) \neq \emptyset$	$E(i,j) \setminus X_j$	$\emptyset$
$X_i \cap E(i,j) \neq \emptyset, X_j \cap E(i,j) = \emptyset$	$\emptyset$	$E(i,j) \setminus X_i$
Any other case	Not defined	Not defined

4.2 Some useful notions and properties

For a partial allocation $X = (X_1, X_2, \dots, X_n)$, we define the following new notions:

- For any two adjacent agents $i \in S$ and $j \in T$, we define $A_{i,j}(X)$ as the set of *available* edges in *E*(*i*, *j*) for *i* and $A_{j,i}(X)$ as the set of *available* edges in *E*(*i*, *j*) for *j*. Formally, we define these sets as specified below in Table 2. Let us assume the *j*-cut configuration of *E*(*i*, *j*) is (C_1, C_2).
- For $i \in [n]$, we define $A_i(X)$ to be her *available set* of edges, i.e., $A_i(X) = \bigcup_{j \neq i, j \in [n]} A_{i,j}(X)$.
- For $i \in [n]$, $U_i(X)$ is the set of all unallocated edges incident to *i*. Note that $A_i(X) \subseteq U_i(X)$.
- For $i \in [n]$, $B_i(X)$ is the set of all *available bundles* for *i*, i.e. $B_i(X) = \{A_{i,j}(X) : j \neq i, j \in [n]\}$.
- For any envied agent $i \in [n]$, we define $S_i(X) \subseteq [n]$ to be her *safe* set, as follows,

$$S_i(X) = \{k \in [n] : k \text{ is non-envied in } X \text{ and } v_i(X_i) \geq v_i(X_k \cup A_i(X))\}.$$

That is, *i* will not envy *k* even if we allocate her whole available set $A_i(X)$ to *k*.
To achieve a complete EFX allocation, we will use an approach that is structurally similar to [22]. We will first find a partial orientation with some nice properties and then allocate the remaining edges to some agent who is not incident to them. Identifying these key nice properties is a non-trivial challenge that we address next.

Key properties We search for a partial allocation $X = (X_1, X_2, \dots, X_n)$ with the following properties:

1. *X* is an EFX orientation.
2. For any two adjacent agents $i \in S$ and $j \in T$, items in *E*(*i*, *j*) must be allocated based on the *j-cut* configuration (C_1, C_2) to either one of their endpoints. By this property, we mean that one of the following cases must happen for *X* (following the Definition 4.1):
   - Either $C_1 \subseteq X_i, C_2 \subseteq X_j$ or $C_2 \subseteq X_i, C_1 \subseteq X_j$.
   - One of the bundles[5] C_1 or C_2 is allocated to agent *i*, and the other bundle is unallocated in *X*.
   - One of the bundles C_1 or C_2 is allocated to agent *j*, and the other bundle is unallocated in *X*.

[5] X_i can also have other items.

- No item in the set $E(i,j)$ is allocated in X.

3. For any agent $i \in [n]$ and a set $B \in B_i(X)$, we have $v_i(X_i) \geq v_i(B)$.
4. For any non-envied agent $i \in [n]$, we have $A_i(X) = \emptyset$.
5. For any non-envied agent $i \in [n]$, we have $v_i(X_i) \geq v_i(U_i(X))$.
6. For any envied agent $i \in [n]$, let j envies i. Then, we have $j \in S_i(X)$.

We are now finally equipped to present our algorithm. We will give a step-by-step procedure to satisfy each key property in the above order. These key properties ensure that there is an easy way to then convert a partial EFX orientation to a complete EFX allocation (see Section 4.7).

As mentioned earlier, for the ease of understanding, we present an example of executing our algorithm step by step on a bipartite multi-graph with additive valuations (see Appendix A).

4.3 Satisfying properties (1)-(3)

We present a greedy algorithm that assigns a set of items to each agent and satisfies the first three properties. It works in the following manner.

Let $S = \{i_1, i_2, \ldots, i_{|S|}\}$ and $T = \{j_1, j_2, \ldots, j_{|T|}\}$ be the two bi-partitions in the given multi-graph G. We fix a picking sequence $\sigma = [i_1, \ldots, i_{|S|}, j_1, \ldots, j_{|T|}]$ that decides the order in which an agent comes and selects her most valuable available bundle. Since the definition of $A_{i,j}(X)$ is dynamic, the set of available bundles for some agents may change after another agent picks her favorite bundle in the picking sequence. Algorithm 1 illustrates this procedure. The properties of this algorithm are further formalized in Lemmas 4.4 and 4.5.

Algorithm 1 Greedy orientation: properties (1)-(3).

Input: A fair division instance on a bipartite multi-graph $G = (S \sqcup T, E)$
Output: A partial EFX orientation satisfying properties (1)-(3)

for $(l \leftarrow 1; l \leq |S|; l+=1)$ **do**
  $k \leftarrow \arg\max_{k \in [n]\setminus\{i_l\}} v_{i_l}(A_{i_l,k}(X))$
  $X_{i_l} \leftarrow A_{i_l,k}(X)$
for $(l \leftarrow 1; l \leq |T|; l+=1)$ **do**
  $k \leftarrow \arg\max_{k \in [n]\setminus\{j_l\}} v_{j_l}(A_{j_l,k}(X))$
  $X_{j_l} \leftarrow A_{j_l,k}(X)$
return $X = (X_1, \cdots, X_n)$

Lemma 4.4 *For a fair division instance on a bipartite multi-graph, the output allocation of Algorithm 1 satisfies properties (1)-(3). Moreover, the algorithm runs in polynomial time if the cut configurations are given as input.*

Proof Let us denote X as the output allocation of Algorithm 1. First, note that X is an orientation since every agent picks items adjacent to them. Next, since every agent selects a bundle greedily, properties (2) and (3) are trivially satisfied. Therefore, what is remaining is to argue that X is EFX. By Observation 2.9, we only need to check the strong envy between adjacent nodes.

Consider an arbitrary agent $i \in S$ and some agent $j \in T$ adjacent to *i*. Note that since *i* precedes *j* in the picking sequence, *i* does not envy *j*. If agent *i* does not pick edges from the set *E*(*i*, *j*), then *j* clearly does not envy *i*. Now, consider the case where agent *i* picks the bundle she prefers from the *j-cut* configuration (C_1, C_2) of *E*(*i*, *j*). Let us assume $v_i(C_1) \geq v_i(C_2)$ and hence *i* picks C_1 (the other case is symmetric). Therefore, C_2 is available for *j* on her turn to pick a bundle in Algorithm 1. Hence, *j* receives a bundle such that $v_j(X_j) \geq v_j(C_2)$. By construction of C_1 and C_2, we know that for every item $g \in X_i = C_1$, we have $v_j(C_2) \geq v_j(C_1 \setminus g)$. Since, $v_j(X_j) \geq v_j(C_2)$, *j* does not strongly envy *i*. Overall, we have shown that *X* is a partial EFX orientation, which completes our proof.

Note that computing each $A_{i,j}(X)$ for every pair of agents *i* and *j* can be done in polynomial time if the cut configurations are given as input. Therefore, since we compute such sets at most *n* times, the overall running time of the algorithm would be polynomial if the cut configurations are given as input. □

Lemma 4.5 *In the output allocation X of Algorithm 1, every envied vertex belongs to the set S.*

Proof We show that agents in *T* are non-envied in *X*. Note that any agent in *S* appears before any agent in *T* in the picking sequence; therefore, they do not envy agents in *T*. Furthermore, agents in *T* do not envy each other and value each other's bundle at zero since they are not connected in the graph skeleton, and our allocation is an orientation. Therefore, any envied vertex (if any) in *X* must belong to the set *S*. □

As we proceed, Lemma 4.5 will continue to hold while we obtain our desired orientation. As demonstrated in the following sections, we will not produce new envied vertices when we modify our allocation *X* to satisfy properties (1)-(6).

4.4 Satisfying property (4)

- Let us now focus on satisfying property (4) that requires $A_i(X) = \emptyset$ for any non-envied agent $i \in [n]$. Let us assume that the output allocation *X* of Algorithm 1 violates property (4). Consider a non-envied agent $i \in [n]$ with $A_i(X) \neq \emptyset$. Therefore, an agent $j \in [n]$ exists such that $A_{i,j}(X) \neq \emptyset$. We will now allocate all of $A_{i,j}(X)$ either to *i* or *j*, depending on the following three possible cases.[6]**Case 1. A set of items in** *E*(*i*, *j*) **is allocated to** *j*: Since $A_{i,j} \neq \emptyset$, by its definition and property (2), no edge in *E*(*i*, *j*) is allocated to *i*. In this case, we can allocate $A_{i,j}(X)$, which is exactly the set $E(i,j) \setminus X_j$ to *i*. Since $X_j \cap E(i,j) \neq \emptyset$, *j* chose the better bundle from the configuration of *E*(*i*, *j*) during Algorithm 1. Also, the set $A_{i,j}(X)$ has value only to agents *i* and *j*; therefore, since agent *i* was non-envied before, the modified allocation remains EFX. One can observe that properties (1)-(3) remain satisfied. Observe that, in this case, *E*(*i*, *j*) will be fully allocated.
- **Case 2. No item in** *E*(*i*, *j*) **is allocated, and** *j* **is non-envied**: Without loss of generality, we can assume $i \in S$ and $j \in T$. Let the partition (C_1, C_2) be the *j-cut* configuration of the set *E*(*i*,*j*). Let us assume $v_i(C_1) \geq v_i(C_2)$ (the other case is symmetric). Observe that

[6] $A_{i,j}(X)$ will be allocated either to *i* or *j*.

by property (3) $v_i(X_i) \geq \max\{v_i(C_1), v_i(C_2)\}$ and $v_j(X_j) \geq \max\{v_j(C_1), v_j(C_2)\}$. Since the X is an orientation, we have that $v_i(X_j) = v_j(X_i) = 0$. We now allocate C_1 to agent *i* and C_2 to agent *j* to obtain,

$$v_i(X_j \cup C_2) = v_i(C_2) \leq v_i(X_i), \text{ and } v_j(X_i \cup C_1) = v_j(C_1) \leq v_j(X_j)$$

Thus, the allocation remains EFX, and all the first three properties are still satisfied. Notice that $E(i, j)$ will be fully allocated in this case as well.

- **Case 3. No item in $E(i, j)$ is allocated and j is envied**: In this case, Lemma 4.5 entails that $j \in S$ and $i \in T$. Let the partition (C_1, C_2) be the *i-cut* configuration of items $E(i, j)$. By property (3) of X, we have $v_j(X_j) \geq \max\{v_j(C_1), v_j(C_2)\}$. Assuming $v_i(C_1) \geq v_i(C_2)$ (the other case is symmetric), we allocate C_1 to agent *i*. Note that agent *j* will not envy *i* and agent *i* remains non-envied in the modified allocation. Hence, the allocation remains EFX, and the properties (1)-(3) are still satisfied.

Formalized protocol (Algorithm 2) to satisfy property (4) along with properties (1)-(3) We repeat the following process as long as there is a non-envied agent $i \in [n]$ who violates property (4). We pick such a violator agent *i*. Then, for every agent $j \neq i$ such that $A_{i,j}(X) \neq \emptyset$, we allocate $A_{i,j}(X)$ according to the cases above. Note that we allocate at least one edge incident to *i* at each step. Therefore, for each agent *i*, this step takes at most $O(m)$ iterations. Then, we repeat. At the end, for any non-envied agent $i \in [n]$, we ensure that $A_i(X) = \emptyset$, thereby satisfying property (4). Moreover, as discussed above, properties (1)-(3) remain satisfied as well. We abuse the notation and call the partial orientation we have built so far (that satisfies properties (1)-(4)) by X.

Algorithm 2 Allocating to non-envied vertices: properties (1)-(3) + property (4).

Input: A partial orientation X that is an output of Algorithm 1 on a bipartite multi-graph G, satisfying properties (1)-(3)
Output: A partial orientation X satisfying properties (1)-(4)

while *there exists a non-envied agent* $i \in [n]$ *such that* $A_i(X) \neq \emptyset$ **do**
    while *there exists an agent* $j \in [n]$ *such that* $A_{i,j}(X) \neq \emptyset$ **do**
        if $X_j \cap E(i, j) \neq \emptyset$ **then**
            $X_i \leftarrow X_i \cup A_{i,j}(X)$
        else if j *is non-envied* **then**
            Without loss of generality, let $i \in S$ and $j \in T$.
            $(C_1, C_2) \leftarrow$ the *j-cut* configuration of $E(i, j)$.
            $C_\ell \leftarrow \arg\max_{C_1, C_2} \{v_i(C_1), v_i(C_2)\}$.
            $X_i \leftarrow X_i \cup C_\ell$
            $X_j \leftarrow X_j \cup C_{3-\ell}$
        else
            $(C_1, C_2) \leftarrow$ the *i-cut* configuration of $E(i, j)$.
            $C_\ell \leftarrow \arg\max_{C_1, C_2} \{v_i(C_1), v_i(C_2)\}$.
            $X_i \leftarrow X_i \cup C_\ell$
return $X = (X_1, \ldots, X_n)$

Claim 4.6 *After satisfying properties (1)-(4), if there exists a pair of agents $k, i \in [n]$ such that $A_{k,i}(X) \neq \emptyset$, then k is an envied vertex, but i is non-envied. Furthermore, $E(k, i) \setminus A_{k,i}(X)$ is allocated to i.*

Proof Note that both *k* and *i* both cannot be envied by Lemma 4.5. Also, since $A_{k,i}(X) \neq \emptyset$, property (4) implies that agent *k* cannot be non-envied. Thus, agent *k* is envied, while agent *i* is non-envied. By property (4), the set $E(k, i) \setminus A_{k,i}(X)$ must be allocated to agent *i*, otherwise $A_{i,k}(X) \neq \emptyset$, which is a contradiction. □

Lemma 4.7 *Algorithm 2 terminates in polynomial time if the cut configurations are given as input.*

Proof For every non-envied agent $i \in [n]$ violating property (4), the outer while loop of Algorithm 2 is executed, and it is clear that if the cut configurations are given as input, then $A_{i,j}(X)$ can be computed in polynomial time for every *i* and *j*. Therefore, for every agent *i* violating property (4), the outer while loop runs in polynomial time. Since there are at most *n* agents violating property (4) and each execution of the outer while loop will fix one of them, the algorithm terminates in polynomial time with the cut configurations given as input. □

4.5 Satisfying property (5)

We now prove that our desired allocation satisfies property (5). Assume that after satisfying properties (1)-(4) there exists a non-envied agent $i \in [n]$ who violates property (5), i.e., $v_i(U_i(X)) > v_i(X_i)$. For every envied agent $j \in S$ adjacent to *i* such that $A_{j,i}(X) \neq \emptyset$, we allocate this set to agent *i* and remove the items from *E*(*i*, *j*) that were previously allocated to *i*. Algorithm 3 and Lemma 4.8 formalize this step.

Algorithm 3 Increase non-envied vertices valuation: properties (1)-(4) + property (5).

Input: A partial orientation X that is an output of Algorithm 2 on a bipartite multi-graph G, satisfying properties (1)-(4)
Output: A partial orientation X satisfying properties (1)-(5)

while *there exists a non-envied vertex* $i \in [n]$ *such that* $v_i(U_i(X)) > v_i(X_i)$ **do**
  for *every envied vertex* $j \in S$ *adjacent to* i *such that* $A_{j,i}(X) \neq \emptyset$ **do**
    Let D be the the set of items currently in $A_{j,i}(X)$
    $X_i \leftarrow X_i \setminus E(i, j)$
    $X_i \leftarrow X_i \cup D$
  if *property (4) is violated* **then**
    Run Algorithm 2
return $X = (X_1, \ldots, X_n)$

Lemma 4.8 *Algorithm 3 terminates and its output allocation satisfies properties (1)-(5). Moreover, it runs in polynomial time if the cut configurations are given as input.*

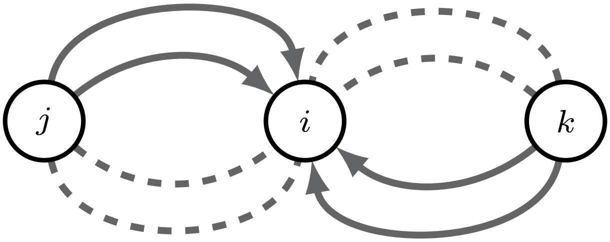

Fig. 6 A demonstration of a non-envied agent *i* violating property (5) after satisfying the first four properties. Agents *j* and *k* are envied vertices adjacent to *i* such that $A_{j,i}(X) \neq \emptyset$ and $A_{k,i}(X) \neq \emptyset$. Dashed edges correspond to unallocated items

Proof Let X and X' denote the input and output orientations of Algorithm 3 respectively. Consider a non-envied agent i violating property (5) in X, i.e., $v_i(U_i(X)) > v_i(X_i)$. By Claim 4.6, every unallocated edge incident to i belongs to $A_{j,i}(X)$ for some envied vertex *j* such that $A_{j,i}(X) \neq \emptyset$ and $E(i,j) \setminus A_{j,i}(X)$ is allocated to *i*. Therefore, Algorithm 3 allocates the set $U_i(X)$ to agent *i*. Figure 6 demonstrates the unallocated edges incident to agent *i*. Observe that agent *i* will not strongly envy any agent in X'. Let *j* be an envied agent incident to *i* with $A_{j,i}(X) \neq \emptyset$. Now, *j* will not envy *i* since we have,

$$v_j(X'_i) = v_j(A_{j,i}(X)) \leq v_j(X_j),$$

where the last inequality follows from property (3) of *X*. Hence, X' remains EFX. Moreover, X' is an orientation, and hence property (1) remains satisfied. Since agent *i* was non-envied in X and remains non-envied in X', property (3) is also clearly satisfied. One can easily observe that property (2) continues to be satisfied, by construction. Therefore, after allocating the set $U_i(X)$ to agent *i* using the for-loop in Algorithm 3, the only property that might be violated is property (4), as it might be the case that agent *i* previously envied some agent $k \in [n]$ in *X*, but since she has been better off, she no longer envies agent *k*, and thus, agent *k* has now become non-envied while $A_k(X) \neq \emptyset$. Therefore, we run Algorithm 2 in these cases to ensure property (4) also remains satisfied.

Since, in every iteration of the while loop of Algorithm 3, the number of vertices violating property (5) decreases by one, the algorithm must eventually terminate. Hence, the output allocation X' satisfies all the first five properties. Moreover, it is clear that for each non-envied agent violating property (5), the inner for-loop takes polynomial time. Therefore, since Algorithm 2 runs in polynomial time, we have that Algorithm 3 terminates in polynomial time given the cut configurations as input.

Note that Claim 4.6 still holds since properties (1)-(4) are satisfied.

4.6 Satisfying property (6)

We now finally focus on satisfying property (6), where for any envied agent $i \in [n]$ who is envied by *j*, it must be that $j \in S_i(X)$.

Algorithm 4 begins by identifying a pair of agents (*i*, *j*) in *X* where *i* is envied by *j* and $j \notin S_i(X)$ and swapping the bundles they possess from the *j-cut* configuration of the set *E*(*i*, *j*). Then, it allocates the set $A_i(X)$ to agent *i* as well. We will show (in Lemma 4.9) that the above procedure satisfies property (6) while maintaining our five previous properties.

Algorithm 4 Safe set: properties (1)-(5)+property (6).

Input: Allocation X satisfying properties (1)-(5)
Output: Allocation X satisfying properties (1)-(6)

1 **while** *there exists an* $i \in [n]$ *who is envied by* $j \notin S_i(X)$ **do**
2 Let the partition (C_1, C_2) be the *j-cut* configuration of the set $E(i, j)$.
3 Swap the bundles C_1 and C_2 between agents i and j.
4 $X_i \leftarrow X_i \cup A_i(X)$
5 **if** *property (4) is violated* **then**
6   Run Algorithm 2
7 **if** *property (5) is violated* **then**
8   Run Algorithm 3
9 **return** $X = (X_1, \ldots, X_n)$

Lemma 4.9 *Algorithm 4 terminates and outputs a partial allocation that satisfies properties (1)-(6). Moreover, it runs in polynomial time if the cut configurations are given as input.*

Proof Consider a single iteration of the while loop of Algorithm 4. Let $i \in [n]$ be the agent violating property (6) and j be the agent who envies i. Let X and X' be the allocations before and after this iteration, respectively. We have,

$$v_i(X'_i) = v_i(X_j \cup A_i(X)) > v_i(X_i),$$

where the last inequality comes from the fact that agent i was an agent violating property (6) previously. Note that, agent i is now better off and does not envy any other agent. During the execution of Lines 2-4 (in Algorithm 4), we swap the bundles C_1 and C_2 between agents i and j, where (C_1, C_2) is the *j-cut* configuration of the set $E(i,j)$ and add $A_i(X)$ to X_i. Therefore, agents i and j are better off and are both non-envied, and the allocation remains EFX. Using a similar argument as in the proof of Lemma 4.8, we have that X' satisfies properties (1)-(3).

Since agent j is now better off, it might be the case that agent j previously envied some agent $k \in [n]$ in X, but since she has been better off, she no longer envies agent k, and thus, agent k has now become non-envied while $A_k(X) \neq \emptyset$. Therefore, we run Algorithm 2 in these cases to ensure property (4) also remains satisfied. Moreover, it might be the case that agent k violates property (5) after executing line 6. Therefore, the second if statement runs Algorithm 3 to ensure property (5) is also satisfied. Note that executing Algorithm 3 maintains the first five properties.

Now, since each iteration of the algorithm decreases the number of envied vertices strictly, we know that Algorithm 4 terminates in at most n iterations. Moreover, given the cut configurations as input, since Algorithms 2 and 3 both terminate in polynomial time, Algorithm 4 also terminates in polynomial time. □

Notice that Claim 4.6 still holds since properties (1)-(4) are satisfied.

4.7 Allocating the remaining items

For a given bipartite multi-graph, we execute Algorithms 1, 2, 3, and 4 (in that order) and reach a desired partial EFX orientation X that satisfies properties (1)-(6). What is remaining is to now make X complete by assigning the unallocated items while maintaining EFX guarantees. We will show that the six properties of X make it easy to do so. By Claim 4.6, we know that the only unallocated edges in X are the set $A_i(X)$ for some envied vertex $i \in [n]$. Let $j \in [n]$ be the agent who envies i. We will allocate $A_i(X)$ to agent j. The procedure is formalized in Algorithm 5. We will prove that this algorithm outputs an EFX allocation.

Algorithm 5 Finalize allocation.

Input: Allocation X satisfying properties (1)-(6)
Output: An EFX allocation
1 **while** *there exists an envied agent* $i \in [n]$ *such that* $A_i(X) \neq \emptyset$ **do**
2 Let j be the agent who envies i.
3 $X_j \leftarrow X_j \cup A_i(X)$
4 **return** $X = (X_1, \ldots, X_n)$

Lemma 4.10 *Algorithm 5 terminates and outputs a complete* EFX *allocation. Moreover, it runs in polynomial time if the cut configurations are given as input.*

Proof To begin with, note that Algorithm 5 terminates in polynomial time if the cut configurations are given as input, since in each iteration at least one edge will be allocated. Moreover, by Claim 4.6, it is clear Algorithm 5 outputs a complete allocation X. Hence, we only need to prove that X is EFX.

Consider an agent $j \in [n]$ who envies agents $i_1, \cdots, i_l$. After the execution of Algorithm 5, this agent has received the set $X_j \cup A_{i_1}(X) \cup \cdots \cup A_{i_l}(X)$. It suffices to show that no agent envies agent j. Observe that $i_1, \cdots, i_l \in S$ and $j \in T$, and also each set $A_{i_p}(X)$ is only valuable to agent $i_p \in S$ and some other agent, called $j_p \in T$. Now, consider an arbitrary agent k. If for every $p \in [l]$, $k \neq j_p$ and $k \neq i_p$, then we have that the set $A_{i_1}(X) \cup \cdots \cup A_{i_l}(X)$ is not valuable for agent k. Since agent j was previously non-envied, we have $v_k(X_j \cup A_{i_1}(X) \cup \cdots \cup A_{i_l}(X)) = v_k(X_j) \leq v_k(X_k)$, meaning that k does not envy j. Now assume $k = j_p$ for some $p \in [l]$. Observe that

$$v_k(A_{i_1}(X) \cup \cdots \cup A_{i_l}(X)) = v_k(A_{i_p}(X)) \leq v_k(U_k(X)),$$

where the last inequality comes from the fact that $A_{i_p}(X) \subseteq U_k(X)$. Moreover, note that $k \in T$ and since the allocation prior to the execution of Algorithm 5 was an orientation, j has not received any item valuable to agent k. Thus,

$$v_k(X_j \cup A_{i_1}(X) \cup \cdots \cup A_{i_l}(X)) = v_k(A_{i_p}(X)) \leq v_k(U_k(X)) \leq v_k(X_k),$$

where the last inequality comes from property (5). Now assume $k = i_p$ for some $p \in [l]$. We have

$$v_k(X_j \cup A_{i_1}(X) \cup \cdots \cup A_{i_l}(X)) = v_k(X_j \cup A_{i_p}(X)) \leq v_k(X_k),$$

where the last inequality comes from property (6). Therefore, in either case, agent k will not envy agent j and the allocation remains EFX. $\square$

Now, we are ready to prove our main result.

Theorem 4.11 *For any fair division instance on a bipartite multi-graph with monotone valuations,* EFX *allocations always exist and can be computed in pseudo-polynomial time. Moreover, if the valuations are cancelable, such allocations can be computed in polynomial time.*

Proof We run Algorithms 1, 2, 3, 4, and 5 in the mentioned order to obtain our EFX allocation. Regarding the running time, by Lemmas 4.4, 4.7, 4.8, 4.9, and 4.10, we can compute our allocation in polynomial time if the configurations are given as input. Note that, using Remark 4.2, we can compute the cut configurations in polynomial time for cancelable valuations and in pseudo-polynomial time for monotone valuations, which completes our proof. □

Now, as a corollary, we obtain the following theorem that says that we can compute orientations that are 1/2-EFX in polynomial time for additive valuations.

Theorem 4.12 *For any fair division instance on a bipartite multi-graph with additive valuations, there always exists an orientation where at least* $\lceil \frac{n}{2} \rceil$ *of agents are* EFX *and the remaining agents are* $\frac{1}{2}$-EFX. *Furthermore, such an orientation can be computed in polynomial time.*

Proof For a given instance, let $S = \{i_1, i_2, \dots, i_{|S|}\}$ and $T = \{j_1, j_2, \dots, j_{|T|}\}$ be two parts of the given bipartite multi-graph. Without loss of generality, we assume assume $|S| \le |T|$. Let X be the partial orientation obtained after executing Algorithms 1 and 2 (in that order). Note that the allocation X satisfies properties (1)-(4).

Using Claim 4.6, we know that the only unallocated bundles in X are between an envied agent i and a non-envied agent j, and the set $E(i,j) \setminus A_{i,j}(X)$ is allocated to j. So, let us consider such a pair of agents (i, j) with an unallocated set C adjacent to them. Note that after satisfying property (3), we have $v_i(C) \le v_i(X_i)$. Since i does not envy j we also have $v_i(X_j) \le v_i(X_i)$. Combining the last inequalities, we have

$$v_i(X_i) \ge \frac{1}{2} v_i(X_j \cup C) \tag{1}$$

We now allocate C to j, and by above inequality, agent i is at least $\frac{1}{2}$-EFX with j's new bundle.

We do this for every such envied and non-envied pair of agents and obtain a full orientation (say, Y) for the given multi-graph. Therefore, every agent in T is EFX-satisfied in Y while, agents in S are $\frac{1}{2}$-EFX satisfied in Y. Since $|S| \le |T|$, the stated claim stands proven. Finally, using Remark 4.2, the cut configurations can be computed in polynomial time for additive valuations, and hence polynomial-time computability of such orientations follow. □

5 Further improvements and limitations

Theorem 4.11 motivates the question of what happens if the graph skeleton contains cycles of odd length. In this section, we extend our techniques to prove that any multi-cycle instance with MMS-feasible valuations admits an EFX allocation (Theorem 5.1). This demonstrates the power of our technique while providing insight and strong hope for potentially proving the existence of EFX on general multi-graphs.

Theorem 5.1 *For any fair division instance on multi-cycles with* MMS*-feasible valuations,* EFX *allocations always exist and can be computed in pseudo-polynomial time.*

Proof To begin with, note that multi-cycles with even length are bipartite graphs, for which Theorem 4.11 proves the existence of EFX allocations. Therefore, we only consider odd-length multi-cycles with at least five vertices.[7] To prove the stated theorem, we distinguish between two main cases, which are further divided into different sub-cases. Here, for every pair of adjacent agents *i* and *j*, we define both the *i-cut* and *j-cut* configurations of the set *E*(*i*, *j*) in the same manner as before (see Definition 4.1).

- **Case 1. There exists a pair of adjacent agents** $i, j \in [n]$ **such that they prefer different bundles in one of the configurations of** *E*(*i*, *j*). Without loss of generality, we assume (C_1, C_2) is the *i-cut* configuration of *E*(*i*, *j*) where $v_i(C_1) \geq v_i(C_2)$ and $v_j(C_2) \geq v_j(C_1)$. In this case, we remove the items of *E*(*i*, *j*) from the graph, which makes the remaining graph a multi-P_n. That is, it becomes a bipartite multi-graph. Since the length of this path is even, we can define the sets *S* and *T* of the vertices in the bipartite graph so that both *i* and *j* belong to the set *T*. Let *X* be the EFX allocation of this multi-bipartite graph using the sets *S* and *T* as described. Therefore, using Lemma 4.5, we know that agents *i* and *j* are both non-envied agents in *X*. Now, we add the edges *E*(*i*, *j*) and allocate C_1 to *i* and C_2 to *j*. First, note that agents *i* and *j* do not envy each other based on the assumption of our case. Furthermore, they remain non-envied, and the resulting allocation is EFX.
- **Case 2. For every pair of adjacent agents** $i, j \in [n]$, **they prefer the same bundle in both configurations of the set** *E*(*i*, *j*). Let us assume j', j, i, i' to be four consecutive agents in the cycle. We remove the agents *i* and *j* and their incident edges from the graph. Then, the remaining graph is a multi-P_{n-2} with even length. Therefore, following the same argument as in the previous case, allocation *X* exists in the remaining graph that is EFX, and both agents j' and i' are non-envied. Let (C_1, C_2), (D_1, D_2), and (E_1, E_2) be the j'*-cut*, *i-cut*, and i'*-cut* for the sets $E(j', j)$, *E*(*j*, *i*), and $E(i, i')$ respectively. Without loss of generality, by the case definition, we assume the following:

$$
\begin{aligned}
v_{j'}(C_1) \geq v_{j'}(C_2), &\text{ and } v_j(C_1) \geq v_j(C_2) \\
v_j(D_1) \geq v_j(D_2), &\text{ and } v_i(D_1) \geq v_i(D_2) \\
v_i(E_1) \geq v_i(E_2), &\text{ and } v_{i'}(E_1) \geq v_{i'}(E_2)
\end{aligned}
$$

We now consider the following cases. (In each case, we allocate the sets $C_1, C_2, D_1, D_2, E_1, E_2$ to agents j', j, i, i'. Note that for each case, one can easily check that agents j', j, i, i' do not strongly envy each other. Furthermore, agents j' and i' remain non-envied.)

  - **Case 2.1.1.** $v_j(C_2 \cup D_2) \geq max(v_j(C_1), v_j(D_1))$ **and** $v_i(D_1 \cup E_2) \geq v_i(E_1)$: We allocate C_1 to j', $C_2 \cup D_2$ to *j*, $D_1 \cup E_2$ to *i*, and E_1 to i'.

[7] Existence of EFX allocation on odd-length multi-cycles with three vertices is known [4] for MMS-feasible valuations.

- **Case 2.1.2.** $v_j(C_2 \cup D_2) \geq max(v_j(C_1), v_j(D_1))$ **and** $v_i(D_1 \cup E_2) < v_i(E_1)$: We allocate C_1 to j', $C_2 \cup D_2$ to j, E_1 to i, and $D_1 \cup E_2$ to i'.
- **Case 2.2.1.** $v_j(C_1) \geq max(v_j(C_2 \cup D_2), v_j(D_1))$ **and** $v_i(D_1 \cup E_2) \geq v_i(E_1)$: We allocate $C_2 \cup D_2$ to j', C_1 to j, $D_1 \cup E_2$ to i, and E_1 to i'.
- **Case 2.2.2.** $v_j(C_1) \geq max(v_j(C_2 \cup D_2), v_j(D_1))$ **and** $v_i(D_1 \cup E_2) < v_i(E_1)$: We allocate $C_2 \cup D_2$ to j', C_1 to j, E_1 to i, and $D_1 \cup E_2$ to i'.
- **Case 2.3.1.** $v_j(D_1) \geq max(v_j(C_2 \cup D_2), v_j(C_1))$ **and** $v_i(D_2 \cup E_2) \geq v_i(E_1)$: We allocate C_1 to j', D_1 to j, $D_2 \cup E_2$ to i, and $C_2 \cup E_1$ to i'.
- **Case 2.3.2.** $v_j(D_1) \geq max(v_j(C_2 \cup D_2), v_j(C_1))$ **and** $v_i(D_2 \cup E_2) < v_i(E_1)$: We allocate C_1 to j', D_1 to j, E_1 to i, and $C_2 \cup D_2 \cup E_2$ to i'.

Therefore, we can find an EFX allocation in multi-cycles of odd length, thereby completing our proof. The running time of this procedure can be concluded from the running time of our algorithm for bipartite multi-graphs (Theorem 4.11). □

Note that our proof works for monotone valuations if the length of the multi-cycle is at least four. Moreover, it is clear to see that it terminates in polynomial time for cancelable valuations. Therefore, we can conclude the following:

Corollary 5.2 *For any fair division instance on multi-cycles with at least four agents and monotone valuations,* EFX *allocations always exist and can be computed in pseudo-polynomial time. Moreover, if the valuations are cancelable, such allocations can be computed in polynomial time.*

Why do our techniques fail for general multi-graphs? Theorems 4.11 and 5.1 make it hopeful that one can adapt/modify our techniques to prove the existence of EFX allocations in general multi-graphs. In our proof, we never had to deal with the case (after Algorithm 1) where there were two envied vertices *i* and *j* in the graph such that no edge from *E*(*i*, *j*) was allocated. This turns out to be the most complicated case for the general multi-graph structure. As a solution concept, one can aim to achieve a partial orientation with $|S_i(X) \cap S_j(X)| \geq 2$ for any adjacent envied vertices *i*, *j*. If this happens, we can let (C_1, C_2) be the *i-cut* configuration of items *E*(*i*, *j*) and then allocate C_1 and C_2 to two different vertices in $S_i(X) \cap S_j(X)$. Also, we believe that we have to use both configurations *i-cut* and *j-cut* for every pair of adjacent vertices *i*, *j* for extending our result to general multi-graphs.

Conjecture: Any fair division instance on a multi-graph admits an EFX allocation.

6 Conclusion

In this work, we study a model that captures the setting where every item is relevant to at most two agents and any two agents can have multiple relevant items in common (represented by multi-graphs). We prove the existence of EFX allocations for fair division instances on bipartite multi-graphs with monotone valuations and multi-cycles with MMS feasible valuations, and that they can be computed in pseudo-polynomial time. Moreover, for bipartite multi-graphs and multi-cycles with length at least four, they can be computed in polynomial time for cancelable valuation functions. An immediate question for future research work is to understand EFX allocations on general multi-graphs, as discussed in Section 5.

The non-existence of EFX orientations on bipartite multi-graphs implies that some degree of *wastefulness* is inherent in EFX allocations for such instances. In this work, we prove the existence of orientations that are EFX for half the agents and 1/2-EFX for the remaining agents for instances represented via a bipartite multi-graph with additive valuations. Another immediate question is, therefore, to improve this factor. Finally, it would be interesting to see what one can say about approximating social welfare or Nash social welfare of EFX allocations to understanding the trade-offs with fairness and efficiencies in multi-graph instances.

Ultimately, we hope that the insights gained from exploring EFX allocations in the multi-graph setting will contribute to advancements in the broader challenge of understanding EFX allocations, in general.

Appendix A: Finding EFX allocation on bipartite multi-graph with additive valuations: an example of the algorithm in Section 4

In this section, we depict the execution of our algorithm for computing an EFX allocation on bipartite multi-graphs on an input instance with additive valuations. We present an example with additive valuations for the ease of understanding.

Figure 7 depicts our input instance. Note that the instance is symmetric, and the numbers on edges specifies the value of that item for both endpoint agents. Note that, in this instance, $q = 2$ and the cut configurations are partitions into two bundles where the size of each bundle is at most one. As we run our algorithm, if we direct an edge towards one of the endpoint agents, it means that the edge is being allocated to that agent. Also, the set of envied vertices are colored blue.

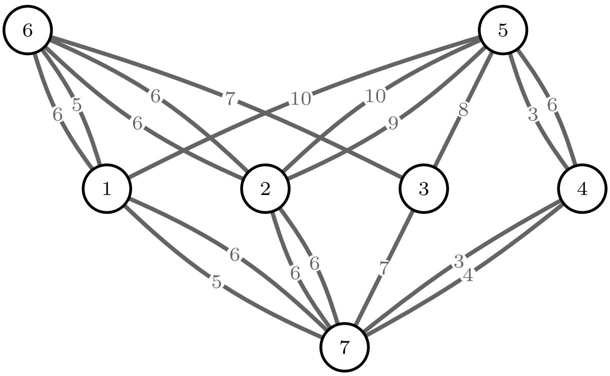

Fig. 7 An instance of a bipartite multi-graph with bi-partitions $S = \{1, 2, 3, 4\}$ and $T = \{5, 6, 7\}$. We will depict the three main steps of our algorithm for finding an EFX allocation using this example. The numbers on the edges depict the value of that edge for both endpoints

We begin by executing Algorithm 1 on our input instance to satisfy properties (1)-(3). The resulting allocation is depicted in Fig. 8. Next, we run Algorithm 2 to satisfy property (4). The resulting allocation is depicted in Fig. 9. It is easy to verify that the said properties are indeed satisfied.

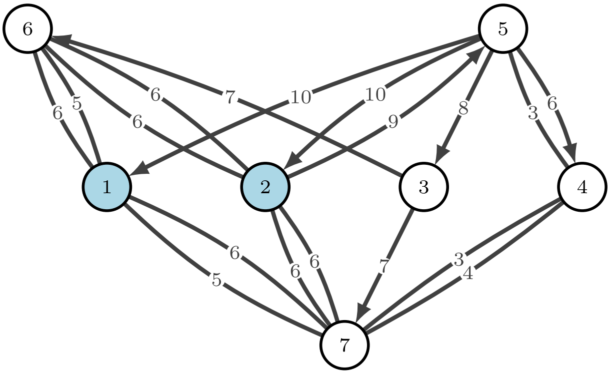

Fig. 8 Partial EFX orientation obtained after executing Algorithm 1 satisfies Properties (1)-(3). The vertices in blue are envied

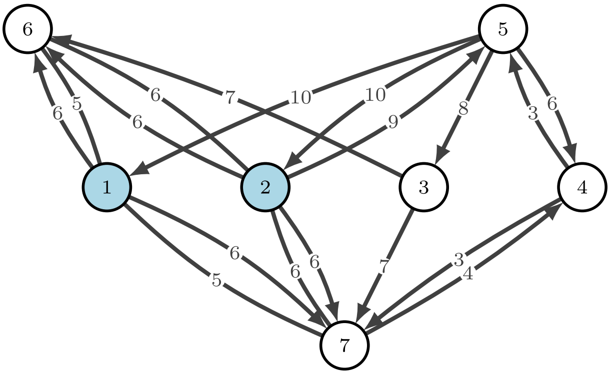

Fig. 9 Partial EFX orientation obtained after executing Algorithm 2 satisfies properties (1)-(4). Note that, Case (1) is executed for the pair (4, 5) of vertices, Case (2) for the pair (7, 4) of vertices, and Case (3) for the pairs (6, 2), (6, 1), (7, 2), (7, 1), and (7, 3) of vertices. The vertices in blue are envied. It is easy to check the correctness of Claim A.1 for non-envied vertices

Observe that, property (5) is automatically *satisfied* in the output (partial) orientation of Algorithm 2 , and hence, we do not need to run Algorithm 3. In fact, for any input instance with additive valuations, if properties (1)-(4) are satisfied, then property (5) is also satisfied. Note that this does not necessarily hold when the valuations are monotone. We formalize this claim as follows:

Claim A.1 *For any fair division instance on a bipartite multi-graph with additive valuations, after satisfying properties (1)-(4) (using Algorithms 1 and 2, executed in that order), we have $v_i(U_i(X)) \leq v_i(X_i)$ for every non-envied vertex $i \in [n]$, i.e., property (5) is also satisfied.*

Proof Consider a non-envied vertex $i \in [n]$ in the output (partial) orientation X (obtained after running Algorithms 1 and 2). By property (4), $A_{i,j}(X) = \emptyset$ for all $j \neq i$. We define the set $R = \{k \in [n] : A_{k,i}(X) \neq \emptyset\}$. Observe that by Claim 4.6, any $k \in R$ must be envied and $U_i(X) = \bigcup_{k \in R} A_{k,i}(X)$. We assume $R \neq \emptyset$; otherwise, the claim holds trivially for i. We now consider an arbitrary agent $k \in R$. Again, Claim 4.6 implies that the set $E(k,i) \setminus A_{k,i}(X)$ is allocated to *i*. Note that agent *i* has received this set either in Algorithm 1 or in the third case of Algorithm 2. Since, in both cases agent *i* had the possibility of choosing between $A_{k,i}(X)$ and $E(k,i) \setminus A_{k,i}(X)$, we have $v_i(E(k,i) \setminus A_{k,i}(X)) \geq v_i(A_{k,i}(X))$. Thus, we can write,

$$\begin{aligned} v_i(U_i(X)) &= v_i\left(\cup_{k\in R} A_{k,i}(X)\right) \\ &\leq v_i\left(\bigcup_{k\in R}(E(k,i) \setminus A_{k,i}(X))\right) \\ &\leq v_i(X_i), \end{aligned}$$

where the last inequality holds due the fact that $E(k,i) \setminus A_{k,i}(X) \subseteq X_i$ for all $k \in R$ and these sets do not intersect with each other. $\square$

Since property (5) is automatically satisfied for additive valuations, we skip Algorithm 3 and directly run Algorithm 4, which satisfies property (6) while maintaining the first five properties. Figure 10 depicts the output allocation at this point.

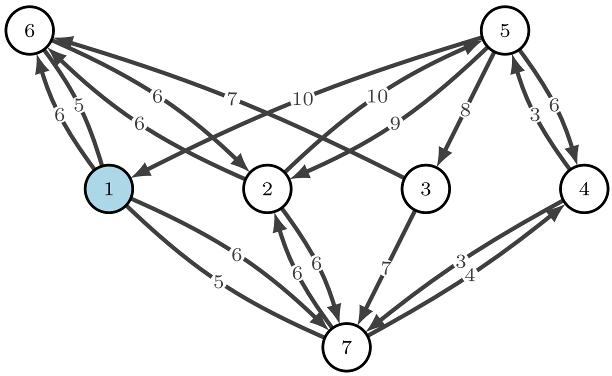

Fig. 10 Partial EFX orientation X obtained after executing Algorithm 4 satisfies properties (1)-(6). Agent 5 envies agent 2 in X and $5 \notin S_2(X)$. Hence, agent 2 gives the item of value ten to agent 5 and receives the item with value nine from agent 5, and then receives $U_2(X)$. Agent 5 envies agent 1 in X and $5 \in S_1(X)$. In the figure, the only unallocated edges are between agents 6 and 1, and 7 and 1. In the last step, since agent 5 envies agent 1, we can allocate both the unallocated edges to agent 5, achieving a complete EFX allocation

Note that, in the output partial EFX orientation X of Algorithm 4, it is only agent 5 who envies agent 1. Therefore, we allocate all the remaining edges to agent 5, yielding a complete EFX allocation.

Acknowledgements A part of the work was done when N. R. was done when she was supported by the Lise Meitner Postdoctoral Fellowship (2023-25), by the Max Planck Institute for Informatics, SIC, Germany. A part of the work was done when M. A. and M. K. were interns at Max Planck Institute for Informatics, SIC, Germany.

Author Contributions Equal contribution from all the co-authors.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Afshinmehr, M., Ansaripour, M., Danaei, A., & Mehlhorn, K. (2024). Approximate EFX and exact tEFX allocations for indivisible chores: Improved algorithms. arxiv:2410.18655
2. Akrami, H., & Rathi, N. (2025). Epistemic EFX allocations exist for monotone valuations. *Proceedings of the AAAI Conference on Artificial Intelligence, 39*, 13520–13528.
3. Akrami, H., Rezvan, R., & Seddighin, M. (2022). An EF2X allocation protocol for restricted additive valuations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI* (pp. 17–23).
4. Akrami, H., Alon, N., Chaudhury, B. R., Garg, J., Mehlhorn, K., & Mehta, R. (2025). EFX: a simpler approach and an (almost) optimal guarantee via rainbow cycle number. *Operations Research, 73*(2), 738–751.
5. Amanatidis, G., Markakis, E., & Ntokos, A. (2020). Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination. *Theoretical Computer Science, 841*, 94–109.
6. Amanatidis, G., Birmpas, G., Filos-Ratsikas, A., Hollender, A., & Voudouris, A. A. (2021). Maximum Nash welfare and other stories about EFX. *Journal of Theoretical Computer Science, 863*, 69–85.
7. Amanatidis, G., Aziz, H., Birmpas, G., Filos-Ratsikas, A., Li, B., Moulin, H., Voudouris, A. A., & Wu, X. (2023). Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence,322*, 103965. ISSN 0004-3702.
8. Ashuri, A., Gkatzelis, V., & Sgouritsa, A. (2025). EF2X exists for four agents. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025*. Philadelphia, PA, USA. AAAI Press.
9. Berendsohn, B. A., Boyadzhiyska, S., & Kozma, L. (2022). Fixed-point cycles and approximate EFX allocations. In S. Szeider, R. Ganian, & A. Silva (Eds.), *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria*, volume 241 of *LIPIcs* (pp. 17:1–17:13). Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
10. Berger, B., Cohen, A., Feldman, M., & Fiat, A. (2022). Almost full EFX exists for four agents. *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, *36*(5), 4826–4833.
11. Bhaskar, U., & Pandit, Y. (2024). EFX allocations on some multi-graph classes. *CoRR*. arxiv:2412.06513.
12. Brams, S. J., & Taylor, A. D. (1996). *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press.
13. Brandt, F., & Procaccia, A. D. (2016). *Handbook of computational social choice*. Cambridge University Press.
14. Budish, E. (2011). The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy, 119*(6), 1061–1103.
15. Budish, E., & Cantillon, E. (2012). The multi-unit assignment problem: Theory and evidence from course allocation at harvard. *American Economic Review, 102*(5), 2237–2271.
16. Caragiannis, I., Kurokawa, D., Moulin, H., Procaccia, A. D., Shah, N., & Wang, J. (2016). The unreasonable fairness of maximum Nash welfare. In *Proceedings of the 2016 ACM conference on economics and computation* (pp. 305–322). ACM.

17. Caragiannis, I., Gravin, N., & Huang, X. (2019). Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)* (pp. 527–545).
18. Caragiannis, I., Garg, J., Rathi, N., Sharma, E., & Varricchio, G. (2023). New fairness concepts for allocating indivisible items. In E. Elkind (Ed.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23* (pp. 2554–2562).
19. Chan, H., Chen, J., Li, B., & Wu, X. (2019). Maximin-aware allocations of indivisible goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 137–143).
20. Chaudhury, B. R., Garg, J., & Mehlhorn, K. (2020). EFX exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)* (pp. 1–19).
21. Chaudhury, B. R., Kavitha, T., Mehlhorn, K., & Sgouritsa, A. (2021). A little charity guarantees almost envy-freeness. *SIAM Journal on Computing, 50*(4), 1336–1358.
22. Christodoulou, G., Fiat, A., Koutsoupias, E., & Sgouritsa, A. (2023). Fair allocation in graphs. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)* (pp. 473–488).
23. Christoforidis, V., & Santorinaios, C. (2024). On the pursuit of EFX for chores: Non-existence and approximations. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence, (IJCAI)* (pp. 2713–2721).
24. Deligkas, A., Eiben, E., Goldsmith, T.-L., & Korchemna, V. (2024). EF1 and EFX orientations. arXiv:2409.13616
25. Dubins, L. E., & Spanier, E. H. (1961). How to cut a cake fairly. *The American Mathematical Monthly, 68*(1P1), 1–17.
26. Etkin, R., Parekh, A., & Tse, D. (2007). Spectrum sharing for unlicensed bands. *IEEE Journal on Selected Areas in Communications, 25*(3), 517–528.
27. Farhadi, A., Hajiaghayi, M., Latifian, M., Seddighin, M., & Yami, H. (2021). Almost envy-freeness, envy-rank, and Nash social welfare matchings. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*, 5355–5362.
28. Foley, D. K. (1967). Resource allocation and the public sector.
29. Garg, J., & Murhekar, A. (2025). Existence of 2-EFX allocations of chores. arxiv:2507.19461
30. Garg, J., Murhekar, A., & Qin, J. (2025). Constant-factor EFX exists for chores. In *Proceedings of the 57th ACM Symposium on Theory of Computing (STOC)*. ISBN 9798400715105.
31. Halpern, D., Procaccia, A. D., Psomas, A., & Shah, N. (2020). Fair division with binary valuations: One rule to rule them all. In *Web and internet economics: 16th International conference, WINE 2020, Beijing, China, December 7–11, 2020, Proceedings 16* (pp. 370–383). Springer.
32. Jahan, S. C., Seddighin, M., Javadi, S. M. S., & Sharifi, M. (2023). Rainbow cycle number and EFX allocations: (almost) closing the gap. In *Proceedings of the thirty-second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China* (pp. 2572–2580). ijcai.org.
33. Lipton, R. J., Markakis, E., Mossel, E., & Saberi, A. (2004). On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM conference on electronic commerce* (pp 125–131). ACM.
34. Mahara, R. (2021). Extension of additive valuations to general valuations on the existence of EFX. In *Proceedings of 29th Annual European Symposium on Algorithms (ESA)* (pp. 66:1–66:15). https://doi.org/10.4230/LIPIcs.ESA.2021.66.
35. Mahara, R. (2025). Existence of fair and efficient allocation of indivisible chores. arxiv:2507.09544
36. Moulin, H. (2004). *Fair division and collective welfare*. MIT press.
37. Plaut, B., & Roughgarden, T. (2020). Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics, 34*(2), 1039–1068.
38. Pratt, J. W., & Zeckhauser, R. J. (1990). The fair and efficient division of the winsor family silver. *Management Science, 36*(11), 1293–1301.
39. Procaccia, A. D. (2020). Technical perspective: An answer to fair division's most enigmatic question. *Communications of the ACM, 63*(4), 118–118.
40. Robertson, J., & Webb, W. (1998). *Cake-cutting algorithms: Be fair if you can*. AK Peters/CRC Press.
41. Sgouritsa, A., & 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 2025*. Detroit, MI, USA.
42. Steinhaus, H. (1948). The problem of fair division. *Econometrica, 16*, 101–104.
43. Stromquist, W. (1980). How to cut a cake fairly. *The American Mathematical Monthly, 87*(8), 640–644.
44. Su, F. E. (1999). Rental harmony: Sperner's lemma in fair division. *The American Mathematical Monthly, 106*(10), 930–942.

45. Vossen, T. (2002). *Fair allocation concepts in air traffic management*. PhD thesis, University of Maryland, College Park.
46. Zeng, J. A., & Mehta, R. (2024). On the structure of envy-free orientations on graphs. arXiv:2404.13527
47. Zhou, S., & Wu, X. (2024). Approximately EFX allocations for indivisible chores. *Artificial Intelligence*, *326*(C). ISSN 0004-3702.
48. Zhou, Y., Wei, T., Li, M., & Li, B. (2024). A complete landscape of EFX allocations on graphs: Goods, chores and mixed Manna. In *Proceedings of the thirty-third International Joint Conference on Artificial Intelligence, IJCAI-24*.

Authors and Affiliations

Mahyar Afshinmehr[1] · Alireza Danaei[2] · Mehrafarin Kazemi[3] · Kurt Mehlhorn[4] · Nidhi Rathi[5]

✉ Mahyar Afshinmehr
mahyar.afshinmehr@cs.ox.ac.uk

Alireza Danaei
alirez11@mit.edu

Mehrafarin Kazemi
mkazemi@mpi-sws.org

Kurt Mehlhorn
mehlhorn@mpi-inf.mpg.de

Nidhi Rathi
n.rathi@uw.edu.pl

1 Department of Computer Science, Linacre College, University of Oxford, Oxford, UK

2 Massachusetts Institute of Technology, Cambridge, MA, USA

3 Max Planck Institute for Software Systems, SIC, Saarbrücken, Germany

4 Max Planck Institute for Informatics, SIC, Saarbrücken, Germany

5 University of Warsaw, Warsaw, Poland