EF1 and EFX Orientations

Argyrios Deligkas[1] , **Eduard Eiben**[1] , **Tiger-Lily Goldsmith**[1] , **Viktoriia Korchemna**[2]
[1]Royal Holloway University of London
[2]TU Wien
{argyrios.deligkas, eduard.eiben, tiger-lily.goldsmith}@rhul.co.uk, vkorchemna@ac.tuwien.ac.at

Abstract

We study the problem of finding fair allocations – EF1 and EFX – of indivisible goods with orientations. In an orientation, every agent gets items from their own predetermined set. For EF1, we show that EF1 orientations always exist when agents have monotone valuations, via a pseudopolynomial-time algorithm. This surprisingly positive result is the main contribution of our paper. We complement this result with a comprehensive set of scenarios where our algorithm, or a slight modification of it, finds an EF1 orientation in polynomial time. For EFX, we focus on the recently proposed graph instances, where every agent corresponds to a vertex on a graph and their allowed set of items consists of the edges incident to their vertex. It was shown that finding an EFX orientation is NP-complete in general. We prove that it remains intractable even when the graph has a vertex cover of size 8, or when we have a multigraph with only 10 vertices. We essentially match these strong negative results with a fixed-parameter tractable algorithm that is virtually the best someone could hope for.

1 Introduction

The allocation of a set of *indivisible* goods to a set of agents in a way that is considered to be *"fair"* is a problem that has been studied since ancient times. Since *envy-free* allocations – no agent prefers the bundle of any other agent over their own – for indivisible goods are not always guaranteed to exist, in recent decades mathematicians, economists, and computer scientists formally studied the problem and have proposed several different fairness solution concepts [Lipton *et al.*, 2004; Bouveret and Lang, 2008; Budish, 2011; Caragiannis *et al.*, 2019b].

Arguably, EF1 and EFX are the two solution concepts that have received the majority of attention in the literature and have created a long stream of work. An allocation is EF1, if it is envy-free up to one good, i.e., any envy from one agent i to some agent j is eliminated by removing a specific item from the bundle of agent j. On the other hand, an allocation is EFX if it is envy-free up to *any* good, i.e., any envy towards agent j is eliminated by removing *any* item from j's bundle. While EFX is a stronger fairness notion, it is unknown whether it always exists; this is one of the main open problems in fair division. On the other hand, EF1 allocations are always guaranteed to exist and in fact, we can efficiently compute such an allocation via the envy-cycle elimination algorithm [Lipton *et al.*, 2004].

However, both EF1 and EFX allow for allocations that can be considered "counterintuitive" in the best case, or *wasteful* in the worst. Consider for example the case where we have two agents, X and Y, and three items, a, b, and c. The valuations of X for a, b, c are $1, 1, 0.2$ respectively, while the valuations of Y are $0, 1, 0$. Observe now that the allocation that gives X item a and Y items b and c is both EFX and EF1. Still, giving item c to agent Y seems rather unreasonable since item c is "irrelevant" to agent Y! Luckily for us, this issue can be fixed by giving item c to X instead. But is such a "fix" always possible? In other words, does a fair allocation always exist under the constraint that every agent gets goods from a restricted set, i.e., a subset of goods that they approve? This is the question we answer in this paper.

Our work is inspired by the recent paper of Christodoulou *et al.* [2023] that studies valuations on graphs. In that model, an instance of the problem is represented via a graph whose vertices correspond to agents with additive utilities and the edges correspond to goods. There, each agent had positive utility only for the goods that corresponded to edges incident to their vertex, i.e., only those goods were *relevant* to them. The value of an agent for every other good, non-incident to their vertex, was zero.

Christodoulou *et al.* studied the existence and complexity of finding EFX allocations and EFX *orientations*. An orientation is an allocation where every agent gets *only* edges adjacent to them, i.e., every edge is "oriented" towards the incident agent that gets it. Christodoulou *et al.* showed something really interesting. They have shown that albeit EFX allocations always exist for this model and they can be computed in polynomial time, EFX orientations fail to exist and in fact, the corresponding problem is NP-complete even for binary, additive and symmetric valuations for the agents.

1.1 Our contribution

Our contribution is twofold: (a) we initiate the study of EF1 orientations; (b) we examine EFX orientations through the lens of parameterized complexity.

Our main result is to prove that an EF1 orientation always exists when the valuations of the agents are monotone! In fact, we prove our result for a more general model than the one from Christodoulou *et al.* [2023], where instead of graphs, we consider hypergraphs, i.e., the goods now correspond to hyperedges. In other words, each agent has a subset of goods that are *relevant* to them. We prove our result algorithmically (Theorem 1). The base of our algorithm is the well-known envy-cycle elimination algorithm [Lipton *et al.*, 2004], although it requires two careful modifications to indeed produce an orientation. The first modification is almost straightforward: every item is allocated to an agent that is incident to it. The second modification is required after we swap the bundles of some agents when we resolve an envy cycle. After the swap, an agent might get goods that are not relevant to them. If this is the case, we *remove* any irrelevant items from all the bundles of the partial allocation and we redistribute them. However, a priori it is not clear whether this procedure will ever terminate. As we prove via a potential argument, the procedure indeed terminates, albeit in pseudo-polynomial time.

Then, we derive polynomial-time bounds for several different valuation classes as direct corollaries of our main theorem, or via a slight modification of the algorithm. Namely, our base algorithm finds an EF1 orientation in polynomial time if every agent has a constant number of relevant items (Corollary 1), or when there exists a constant number of "local" item-types (Corollary 2). In addition, via straightforward modifications of the base algorithm, we can efficiently compute EF1 orientations for identical valuations (Theorem 2), or when the relevant items of the agents form *laminar* sets (Corollary 3).

For EFX orientations, we begin by showing two strong negative results. Firstly, we show that it is NP-complete to decide whether an EFX orientation exists even on graphs with vertex cover of size 10, even when the valuations are additive and symmetric (Theorem 3). This result rules out the possibility of fixed-parameter algorithms for a large number of graph parameters. Furthermore, we show that if we consider multigraphs instead of graphs, i.e., we allow parallel edges, finding an EFX orientation is NP-hard even when we have 8 agents with symmetric and additive valuations (Theorem 4). We complement these intractability results with a fixed parameter algorithm, for which the analysis is rather involved, parameterized by the slim tree-cut width of the underlying graph; this is essentially the best result someone could hope for, given our previous results.

Due to space constraints, some details, marked with ⋆, are omitted and are available in the supplementary material.

1.2 Related Work

The recent survey by Amanatidis *et al.* [2023] provides a comprehensive coverage of work on fair division of indivisible goods. In this section, we direct the reader to some other papers, in particular, that study EFX or EF1 and restrict the instance in different ways.

As aforementioned, the question of whether EFX always exists is a well-known open question in Fair Division. Currently, we have that Plaut and Roughgarden [2020] prove EFX always exists for 2 agents. For 3 agents already this question is much harder, Chaudhury *et al.* [2024] prove that EFX exists for 3 agents but with additive valuations, and recently Akrami *et al.* [2024] prove a result for 3 agents where only 1 of these agents requires additive valuations (and the other 2 agents have MMS-feasible valuations). The paper by Goldberg *et al.* [2023] studies the intractability of EFX with just two agents. They find that even with a small instance like this, it quickly becomes intractable as the valuations become more general – namely computing an EFX allocation for two identical agents with submodular valuations is PLS-hard[1]. However, they propose an intuitive greedy algorithm for EFX allocations for weakly well-layered valuations; a class of valuations which they introduce. An example of relaxation of EFX that has been studied is EFkX, envy freeness up to k goods, Akrami *et al.* [2022] study EF2X and prove existence in some restricted settings. A recent paper by Zhou *et al.* [2024] studies EFX allocations in the mixed setting on graphs, where agents only have valuations for edges adjacent to them and these can be positive or negative. They treat orientations as a special case of their problem and show that deciding if an EFX orientation exists is NP-complete. The paper by Payan *et al.* [2023] also studies graph restrictions but these are subtly different to that of Christodoulou *et al.*. In this model, edges are not items but instead, they are where EFX (/other fairness notions) must apply, intuitively this aims to capture a model where we want envy freeness between an agent and some of their neighbors. Some studies look at EFX where we (may) decide to leave some items unallocated. We refer to this as EFX with charity, [Caragiannis *et al.*, 2019a; Chaudhury *et al.*, 2021], where not all items are allocated and these leftover items are said to be "given away to charity". Moreover, Caragiannis *et al.* [2019b] introduces EFX0 and Kyropoulou *et al.* [2020] studies this. An allocation satisfies EFX0 if for one agent i they are not envious of any other agent j's bundle after removing any item for which agent i doesn't have a positive value. Moreover, another model which may be of interest is that of connected fair division - originally introduced by Bouveret *et al.* [2017] and more recently Deligkas *et al.* [2021] studies this under the lens of parameterized complexity - in which there are some items which cannot be separated i.e. have some *connectivity* constraints.

2 Preliminaries

Throughout the paper we consider $A = \{a_1, a_2, \ldots, a_m\}$ to be a set of indivisible items and $N = \{1, 2, \ldots, n\}$ to be a set of agents. An allocation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a partition of the items into n (possibly empty) sets which we refer to as bundles. Thus, $\pi_i \cap \pi_j = \emptyset$ for every $i \neq j$ and $\bigcup_{i \in N} \pi_i = A$, where the bundle π_i is allocated to agent i. For an item $a \in A$, we denote by $\pi(a)$ the agent who receives item a in the allocation π. We refer to an allocation of a subset of items as a partial allocation.

Every agent $i \in N$ has a *valuation* function $\mathcal{V}_i$ that assigns a value $\mathcal{V}_i(S)$ for every subset $S \subseteq A$, where $\mathcal{V}_i(\emptyset) = 0$. $\mathcal{V}_i$ is *non-negative* if for all $S \subseteq A$ it holds $\mathcal{V}_i(S) \geq 0$; $\mathcal{V}_i$ is *monotone* if for all $S' \subset S$ it holds $\mathcal{V}_i(S') \leq \mathcal{V}_i(S)$; $\mathcal{V}_i$

[1]For the definition of PLS see Johnson *et al.* [1988].

is *additive* if there exist non-negative values $v_{i1}, v_{i2}, \ldots, v_{im}$ such that for every $S \subseteq A$ it holds $\mathcal{V}_i(S) = \sum_{j \in S} v_{ij}$.

For an allocation π and two agents $i, j \in N$, we say that i *envies* j or alternatively that there is *envy from* i *towards* j if $\mathcal{V}_i(\pi_j) > \mathcal{V}_i(\pi_i)$. An allocation is *fair* if envy can be eliminated in some particular way.

Definition 1 (EF1)**.** *An allocation π is envy-free up to one item (EF1), if for every pair of agents $i, j \in N$ there* exists *an item $a \in \pi_j$ such that $\mathcal{V}_i(\pi_i) \geq \mathcal{V}_i(\pi_j \setminus \{a\})$.*

Definition 2 (EFX)**.** *An allocation π is envy-free up to any item (EFX), if for every pair of agents $i, j \in N$ and* every *$a \in \pi_j$ it holds that $\mathcal{V}_i(\pi_i) \geq \mathcal{V}_i(\pi_j \setminus \{a\})$.*

Relevant items. We say that an item a is *relevant* for an agent i if there is a set S of items such that $\mathcal{V}_i(S \setminus a) < \mathcal{V}_i(S)$. For every agent i, we will denote the set of items relevant to i by A_i. Similarly, for every item a, we let $N_a = \{i \in N \mid a \in A_i\}$ be the set of agents to which a is relevant, we will also call N_a the *agent list* of a. We say that items a and b belong to the same *group* if $N_a = N_b$. Throughout the paper we assume that the union of all relevant sets is the set of items or, in other words, every item is relevant for at least one agent.

Orientations. Using relevant items, we can define a subset of all possible allocations, that we call *orientations*. Formally, an allocation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is called an *orientation* if $\pi_i \subseteq A_i$ for all $i \in N$. In other words, in an orientation, the bundle of an agent contains *only* relevant items. For $\phi \in \{\text{EF1}, \text{EFX}\}$, we say that an allocation π is a ϕ orientation if it is an orientation and in addition, it satisfies the corresponding fairness definition.

Parameterized Complexity. An instance of a parameterized problem $Q \subseteq \Sigma \times \mathbb{N}$, where Σ is fixed and finite alphabet, is a pair (I, k), where I is an input of the problem and k is a *parameter*. The ultimate goal of parameterized algorithmics is to confine the exponential explosion in the running time of an algorithm for some NP-hard problem to the parameter and not to the instance size. The best possible outcome here is the so-called *fixed-parameter* algorithm with running time $f(k) \cdot |I|^{\mathcal{O}(1)}$ for any computable function f. That is, for every fixed value of the parameter, we have a polynomial time algorithm where the degree of the polynomial is independent of the parameter. For a more comprehensive introduction to parameterized complexity, we refer to the monograph of Cygan *et al.* [2015].

3 EF1 Orientations for Monotone Valuations

In this section we establish the existence of EF1 orientations when agents have monotone valuations via the construction of a pseudopolynomial-time algorithm and we identify several sub-classes of valuation functions for the agents where our algorithm, or a slight modification of it, finds an EF1 orientation in polynomial time.

Let $\mathcal{V}_i$ be a valuation function of an agent. The maximal range of $\mathcal{V}_i$ is the number of different values of a bundle assigned by $\mathcal{V}_i$, i.e., $|\{\mathcal{V}_i(S) \mid S \subseteq A\}|$.

Theorem 1. *When agents have monotone valuations, an EF1 orientation always exists and can be computed in time $\mathcal{O}\left(mn^3 r\right)$ where r is the maximal range size of $\mathcal{V}_i$, $i \in N$.*

Algorithm 1 EF1 orientations for monotone valuations

Input: Set of items A, set of agents N, valuations $\mathcal{V}_i$, sets of relevant items A_i, $i \in N$, agent lists N_a, $a \in A$
Output: EF1 orientation π

Let $\pi = (\pi_1, \ldots, \pi_n)$ such that $\pi_i = \emptyset$ for all $i \in N$.
Let $G_\pi = (N, \emptyset)$ be a directed graph (an "envy-graph" of π)
while $\exists$ item a that is not in any π_i **do**
  Let $i \in N_a$ be an agent such that i is a source-vertex in $G[N_a]$.
  Let $\pi_i = \pi_i \cup \{a\}$
  for $j \in N_a \setminus \{i\}$ **do**
    if $\mathcal{V}_j(\pi_j) < \mathcal{V}_j(\pi_i)$ **then**
      Add the edge from j to i if it does not exist.
    end if
  end for
  Call Algorithm 2 with π and G_π to eliminate all cycles in G_π.
end while
return π

Algorithm 2 Eliminating cycles in the envy-graph

Input: partial allocation π with an envy-graph G_π
Output: partial allocation π' with $\mathcal{V}_i(\pi'_i) \geq \mathcal{V}_i(\pi_i), i \in N$, such that $G_{\pi'}$ does not contain directed cycles

while $\exists$ a directed cycle $C = (i_1, i_2, \ldots, i_\ell)$ in G_π **do**
  for all $j \in [\ell - 1]$ **do**
    Let $\pi'_{i_j} = \pi_{i_{j+1}} \cap A_{i_j}$
  end for
  $\pi'_{i_\ell} = \pi_{i_0} \cap A_{i_\ell}$
  Let $\pi = \pi'$
  Recompute G_π
end while
return π

Proof. A full pseudo-code of the algorithm is given in Algorithm 1. The algorithm is inspired by the envy-cycle elimination algorithm by Lipton *et al.* [2004]. Similarly to that algorithm, we are computing the allocation π iteratively starting from an empty (partial) allocation keeping an envy graph G_π, which is a graph whose vertex set is precisely the set of agents N and there is a directed edge from i to j if in the allocation π the agent i envies the agent j. In addition, we will be preserving that π is an EF1 (partial) allocation such that $\pi_i \subseteq A_i$ for all $i \in N$. For the ease of the presentation of the proof, we will say that a pair of agents i, j satisfy EF1-property if $\mathcal{V}_i(\pi_j) \leq \mathcal{V}_i(\pi_i)$ or there exists $a \in \pi_j$ such that $\mathcal{V}_i(\pi_j \setminus \{a\}) \leq \mathcal{V}_i(\pi_i)$.

While the algorithm by Lipton *et al.* greedily picks any source-vertex i in G_π (that is a vertex without any edge pointing towards it; so no agent in N envies agent i for its bundle in π) and allocates to i an arbitrary item, this would not work for us, as the remaining items might not be relevant for the source vertices of G_π. Instead, we first pick an unassigned item a that needs to be assigned and give it to an agent i that is a source in the subgraph of G_π induced by the agents in the set N_a that are allowed to receive that item. Hence, the item

a is irrelevant for any agent j that envies i before allocating the item a. Therefore, these steps preserve both that $\pi_i \subseteq A_i$ for all $i \in N$ and that for all $i, j \in N$ the pair i, j satisfies EF1-property.

Furthermore, in order to always find a source-vertex in this induced subgraph of the envy graph, we also need to be able to eliminate cycles in G_π. An algorithm for eliminating all cycles in G_π is described in Algorithm 2. Let $C = (i_1, i_2, \ldots, i_\ell)$ be a cycle such that the agent i_j envies the bundle of the agent i_{j+1} for $j \in [\ell - 1]$ and the agent i_ℓ envies the bundle of the agent i_1. Similarly to Lipton *et al.*, we shift the bundles in the opposite direction of the cycle, which is known to eliminate the envy on the cycle and does not create any new envy. However, after we execute this shift of bundles, the bundle of an agent i on the cycle could contain some items that are not in the set A_i of their relevant items. So we remove from the bundle of the agent i any item that is not relevant for them. Note that this does not change the valuation of i for its bundle. Let us denote by π the allocation before the cycle-elimination step and by π' the allocation after the cycle-elimination step. As we already discussed, for all $i \in N$ we have $\mathcal{V}_i(\pi'_i) \geq \mathcal{V}_i(\pi_i)$ (the value either increased or the bundle did not change). Now let $i, j \in N$. We know that for all $j \in N$, the pair i, j satisfies EF1-property in π. It follows that if $\pi'_j = \pi_j$, then i, j satisfies EF1-property as well. Else $\pi'_j \subseteq \pi_k$ for some $k \in N$. It follows from monotonicity of the valuation that $\mathcal{V}_i(\pi'_j) \leq \mathcal{V}_i(\pi_k)$ and $\mathcal{V}_i(\pi'_j \setminus a) \leq \mathcal{V}_i(\pi_k \setminus a)$ for all $a \in \pi_k$ and so the pair i, j also satisfies the EF1-property in π'.

It follows from the above discussion and from the fact that Algorithm 1 stops only when all items have been allocated that whenever Algorithm 1 stops it returns an EF1 orientation. It only remains to show that the algorithm terminates.

Let us consider the vector $W^\pi = (\mathcal{V}_1(\pi_1), \mathcal{V}_2(\pi_2)), \ldots, \mathcal{V}_n(\pi_n))$. By definition, each coordinate W^π_i of the vector W^π has at most $r \leq 2^m$ many possible values. In addition, in each cycle-elimination step all the coordinates corresponding to the agents on the cycle strictly increase and the remaining coordinates of W^π do not change. Similarly, adding item a to agent i on line 5 of Algorithm 1 does not decrease any of the coordinates because of the monotonicity of the valuations. Since every coordinate can increase its value only $r - 1$ times, it follows that there are less than $n \cdot r$ many cycle elimination steps in total, each can be executed in $\mathcal{O}(nm)$ time. Between any two cycle-eliminations we can only add less than m items after each addition of an item, we need to update G_π and check whether a cycle is created, which can be easily done in $\mathcal{O}(|V(G_\pi)| + |E(G_\pi)|) = \mathcal{O}(n^2)$ time. Putting everything together, Algorithm 1 runs in $\mathcal{O}(nr \cdot (nm + mn^2)) = \mathcal{O}(mn^3 r))$ time. □

Note that if the number of relevant items for each agent is constant and bounded by some $\ell \in \mathbb{N}$, then the number of possible bundles for each agent and hence the range r of its valuation function $\mathcal{V}_i$ is bounded by 2^ℓ. Therefore, Theorem 1 immediately implies the following corollary.

Corollary 1. *For monotone valuations, computing an EF1 orientation is fixed-parameter tractable parameterized by the number of relevant items per agent, l and can be computed in time $\mathcal{O}(mn^3 \cdot 2^l)$.*

3.1 Local item-types

While in Corollary 1, the number of relevant items per agent is small, our algorithm provided in Theorem 1 can be applied to compute an EF1 orientation in polynomial time in much more general settings. For instance, if the range of each valuation has size at most m^d for some constant d, the running time is upper-bounded by $\mathcal{O}(m^{d+1}n^3)$.

One natural example of such valuations is as follows. Assume that each agent subdivides all items relevant to them into d groups (which we will refer to as *local item-types*) and only distinguishes items that are in different groups. In this case, the valuation each agent has for their bundle depends only on the number of received items from each local item-type. Since each of the d groups contains at most m items, there are at most m^d possibilities for their value.

Corollary 2. *For monotone valuations, computing an EF1 orientation is XP parameterized by d, the number of local item-types per agent, and can be computed in time $\mathcal{O}(m^{d+1}n^3)$.*

In general, when the range size of $\mathcal{V}_i$ is unbounded, we still can distinguish some settings for which the described algorithm (or its slight modification) is polynomial.

3.2 Identical valuations

While strictly speaking, identical valuations would mean that all items have to be relevant for all agents, we relax this notion slightly to better fit with the intended meaning of the relevant items as a restriction of the item an agent is allowed to receive.

Definition 3. *A set of agents have* identical *valuations if there exists a function $\mathcal{V}$ such that for every agent i their valuation function is defined by $\mathcal{V}_i(B) = \mathcal{V}(B \cap A_i)$ for every $B \subseteq S$.*

Similarly to the standard setting where agents with identical valuations cannot create envy cycles, we can show that the same is the case even with this relaxed definition of identical valuations and we obtain the following theorem.

Theorem 2. *An EF1 orientation can be computed in linear time when agents have identical monotone valuations.*

Proof. We will show that it is not possible to have a cycle in the envy graph. That is that when we run Algorithm 1, then whenever Algorithm 2 is called as a subroutine, the condition on line 1 in Algorithm 2 is always false and it returns the same partial allocation.

For the sake of a contradiction, assume that $C = (i_1, i_2, \ldots, i_\ell)$ is a cycle in the envy-graph for some partial allocation π such that $\pi_i \subseteq A_i$. That is the agent i_j envies the agent i_{j+1} for all $j \in [\ell - 1]$ and the agent i_ℓ envies the agent i_1. Let $\mathcal{V}$ be the function such that for all $i \in N$ and for all $B \subseteq A$ we have $\mathcal{V}_\rangle(B) = \mathcal{V}(B \cap A_i)$. Since, $\pi_i \subseteq A_i$ for all $i \in N$ it follows that $\mathcal{V}_i(\pi_i) = \mathcal{V}(\pi_i)$ for all $i \in N$ and $\mathcal{V}_i(\pi_j) = \mathcal{V}(\pi_j \cap A_i)$ and by monotonicity $\mathcal{V}_i(\pi_j) \leq \mathcal{V}(\pi_j)$. It follows that

$$\mathcal{V}(\pi_{i_1}) < \mathcal{V}(\pi_{i_{j+1}}) < \cdots < \mathcal{V}(\pi_{i_\ell}) < \mathcal{V}(\pi_{i_1}),$$

which is a contradiction. Therefore, Algorithm 1 only has to assign every item a once to a source-vertex in $G_\pi[N_a]$, without ever running Algorithm 2 which takes linear time $\mathcal{O}(mn)$. □

3.3 Laminar agent lists

The final setting for which we get a polynomial time algorithm is when items are arranged in some kind of hierarchical structure, where the most common items can be assigned to any agent and then we have more and more specialized items that only a smaller and smaller group of agents can get. One can think about it like this: agents need to undergo some training and which items you are allowed to receive depends on your specialization and on the amount of training you already received.

Definition 4. *We say that agent lists N_a, $a \in A$, are **laminar** if for any two items a_1 and a_2 it holds that either $N_{a_1} \cap N_{a_2} = \emptyset$ or one of the sets is a subset of another, i.e. $N_{a_1} \subseteq N_{a_2}$ or $N_{a_2} \subseteq N_{a_1}$*

Corollary 3. *For the monotone valuations with laminar agent lists, an EF1 orientation can be computed in time $\mathcal{O}\left(m^2 n\right)$.*

Proof. Recall that the algorithm described in the proof of Theorem 1 picks undistributed items in random order. Here, instead of this, if the agent lists are laminar, we order the items in a tuple $(a_1, \ldots, a_m)$ so that either $N_j \cap N_k = \emptyset$ or $N_k \subseteq N_j$ whenever $1 \leq j < k \leq m$. In the algorithm, we will pick items exactly in this order. Consider the moment when a new item a is allocated. Then shifting items along the cycle in G_a will never result in allocating illegal items. Indeed, if some item b was moved to agent $i \in N_a$, then $N_a \cap N_b \neq \emptyset$. Since b was allocated before a, we conclude that $N_a \subseteq N_b$ and hence $i \in N_b$.

Therefore, each item is moved from not distributed to distributed precisely once, and potential cycle elimination afterwards takes time at most $\mathcal{O}(mn)$, so the EF1 orientation is computed in time at most $\mathcal{O}\left(m^2 n\right)$. □

4 EFX Orientations

The paper by Christodoulou *et al.* [2023] proves that it is NP-complete to decide if an EFX orientation exists. We strengthen this result, by showing it is NP-hard even when the graph has constant vertex cover.

Theorem 3. *Deciding whether an EFX orientation exists on graph G is weakly NP-hard even when G has a vertex cover of constant size.*

Proof. We show a reduction from the NP-complete problem PARTITION. An instance of partition consists of a multiset S of positive integers. Let T be the number of elements in S. Let $\sum_{t=1}^{T} S_t = 2B$. Given S we want to decide if the elements can be divided into two subsets S_1 and S_2 such that the sum of elements in S_1 is equal to that of S_2. Given an instance of PARTITION, we will construct a graph $G = (V, E)$. We start by constructing a bipartite graph such that there is a vertex (i.e. an agent) for every element in S on one side, call them $x_1, \cdots, x_T$, and two additional vertices i and j on the other. Now we will create the edges, where $|E| = m$, the number of items and weights on edges are the valuation of that item for the agents on both endpoints. For all vertices $x_v \in \{x_1, x_2 ..., x_T\}$ we create an edge (i, x_v) of weight S_v and an edge (j, x_v) of weight S_v. We create two copies of gadget X, we will call these X_1 and X_2. Gadget X is a clique on 4 vertices based on Example 1. in [Christodoulou *et al.*, 2023] that has no EFX orientation on its own. Let the vertices in the gadget X_h be X_{h1} to X_{h4} . Edges (X_{h1}, X_{h2}) and (X_{h3}, X_{h4}) have weight B. All other edges in X_h have weight 1. Finally, we create edges (i, X_{11}) and (j, X_{21}) of weight B. This completes the construction, see Figure 1.

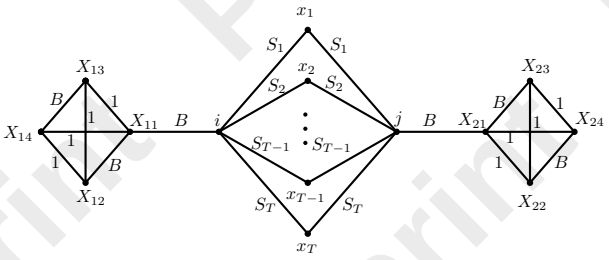

Figure 1: The construction used in the proof of Theorem 3.

One can now show that in order to get an EFX orientation, the edges (i, X_{11}) and (j, X_{21}) have to be allocated to X_{11} and X_{21}, respectively. In addition, the only way to satisfy all the remaining agents in the gadgets X_1 and X_2, both X_{11} and X_{21} need to receive additional edge of value 1. Hence in order for both i and j to be satisfied, they each need to value their bundle at least B. Since to achieve this both i and j need to receive more than just one edge, each of agents x_k for $k \in [T]$ has to receive at least one of its incident edges. Therefore, the only way for the graph G to admit EFX orientation is if the elements associated with the edges allocated to i and the elements associated with the edges allocated to j form a partition of S. □

Theorem 4 (⋆)**.** *Deciding whether an EFX orientation exists on a multigraph G is weakly NP-hard even when G has a constant number of vertices.*

Proof Sketch. We begin with a construction very similar to that of Theorem 3. The only difference is that instead of creating a set of vertices $x_1 \cdots x_T$ we will create T many edges between vertices i and j, see Figure 2. □

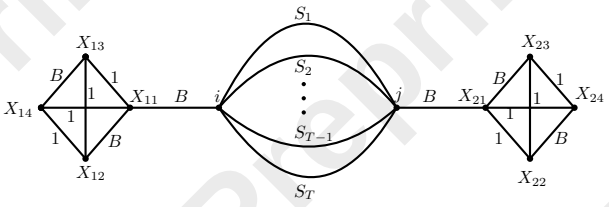

Figure 2: The construction used in the proof of Theorem 4.

4.1 Slim tree-cut width and the FPT algorithm

Examining the hardness of Christodoulou *et al.*, we can see that their reduction can be made to work on graphs with constant maximum degree. In combination with Theorem 3, this shows that efficient algorithms for deciding the existence of EFX orientations are unlikely already for very restricted settings. In this section, we show an efficient algorithm for a setting that is basically on the limit of tractability from the point of view of parameterized complexity (in a sense, that more general parameters studied so far would contain some of the hard instances).

As follows from our reduction, deciding whether EFX orientation exists in the graph setting is hard, even when the underlying graph has constant vertex cover and tree-cut width. In particular, this rules out most of the vertex-separator based parameters. However, we show that the recently introduced parameter slim tree-cut width (also equivalent to super edge-cut width, see [Ganian and Korchemna, 2024]) allows us to achieve tractability.

For simplicity, here we work with super edge-cut width. For a graph G and a spanning tree T of G, let the *local feedback edge set* at $v \in V$ be $E^{G,T}_{\text{loc}}(v) = \{uw \in E(G) \setminus E(T) \mid$ the unique path between u and w in T contains $v\}$.

Definition 5. *The edge-cut width of the pair (G,T) is* $\text{ecw}(G,T) = 1+\max_{v\in V} |E^{G,T}_{\text{loc}}(v)|$*, and the edge-cut width of G (denoted* $\text{ecw}(G)$*) is the smallest edge-cut width among all possible spanning trees T of G.*

If in the last definition, we allow to choose the spanning tree in any connected supergraph of G, this leads to the notion of *super edge-cut width* (denoted by $\text{sec}(G)$):

$$\text{sec}(G) = \min\{\text{ecw}(H,T) \mid H \supseteq G, T - \text{spanning tree of } H\}.$$

Super edge-cut width is a strictly more general parameter than degree+treewidth and feedback edge number, but it is more restrictive than tree-cut width [Ganian and Korchemna, 2024].

Theorem 5 ($\star$)**.** *Deciding whether an EFX orientation exists for a given graph G is fixed-parameter tractable if parameterized by the slim tree-cut width of G.*

4.2 Proof of Theorem 5

Let H be the supergraph of G with the spanning tree T such that $\text{sec}(G) = \text{ecw}(H,T) = k$. For the rest of the proof, we root T in some arbitrary fixed vertex. We begin by recalling some basic properties of the super edge-cut width, in particular, we adapt the related notion of the boundary of a vertex, firstly introduced in [Ganian and Korchemna, 2021] for a weaker parameter.

For $v \in V(T)$, let T_v be the subtree of T rooted at v, let $V_v = V(G \cap T_v)$, and let $\bar{V}_v = N_G(V_v) \cup V_v$. We define the *boundary* $\delta(v)$ of v to be the set of endpoints of all edges in G with precisely one endpoint in V_v (observe that the boundary can never have a size of 1). A vertex v of T is called *closed* if $|\delta(v)| \leq 2$ and *open* otherwise.

Observation 1 ([Ganian and Korchemna, 2021])**.** *Let v be a vertex of T. Then:*

1. *$\delta(w) = \{v, w\}$ for every closed child w of v in T, and vw is the only edge between V_w and $V \setminus V_w$ in G.*
2. *$|\delta(v)| \leq 2k + 2$.*
3. *Let $\{v_i | i \in [t]\}$ be the set of all open children of v in T. Then $t \leq 2k$ and $\delta(v) \subseteq \cup_{i=1}^{t} \delta(v_i) \cup \{v\} \cup N_G(v)$.*

We can now define the records that will be used in our dynamic program. Intuitively, these records will be computed in a leaf-to-root fashion and will store at each vertex v information about possible EFX orientations for the subtree of T rooted at v.

Let R be a binary relation on $\delta(v)$, and S a subset of $\delta(v)$.

Definition 6. *We say that (R,S) is a* record *at vertex v if there exists an orientation D of G restricted to vertices in $\bar{V}_v$ and edges with at least one endpoint in V_v such that:*

1. *The partial allocation defined by D is EFX between any two vertices in V_v.*
2. *R is the set of all arcs of D that have precisely one endpoint in V_v.*
3. *For every $w \in \delta(v)$, if there is envy from some vertex $u \in V_v$ towards w in D then $w \in S$.*
4. *The in-degree of every $w \in S$ in D is equal to one.*

We call such a digraph D a *partial solution* at v. We say that D is a *witness* of the record (R,S). Denote the set of all records for v by $\mathcal{R}(v)$, then $|\mathcal{R}(v)| \leq 2^{\mathcal{O}(k^2)}$.

Intuitively, the set S in a record is intended to capture all the vertices of $\delta(v)$ towards which there can be envy in the resulting solution. This is why we require all the vertices in S to have the in-degree of one. Otherwise, there would be a possibility to remove items without changing the envy: note that the only relevant item for both agents is their shared edge.

Note that if v_i is a closed child of v, then by Observation 1, $\mathcal{R}(v_i)$ can contain only the records $(\{v_iv\}, \emptyset)$, $(\{v_iv\}, \{v\})$, $(\{vv_i\}, \emptyset)$ and $(\{vv_i\}, \{v_i\})$. The root r of T contains at most one record $(\emptyset, \emptyset)$, which happens if and only if the instance is a YES instance (otherwise $\mathcal{R}(r) = \emptyset$).

Observation 2 ($\star$)**.** *Let v_i be a closed child of v.*

- *If $(\{v_iv\}, \emptyset) \in \mathcal{R}(v_i)$, then $(\{v_iv\}, \{v\}) \in \mathcal{R}(v_i)$,*
- *if $(\{vv_i\}, \{v_i\}) \in \mathcal{R}(v_i)$, then $(\{vv_i\}, \emptyset) \in \mathcal{R}(v_i)$.*

Lemma 6 ($\star$)**.** *Let $v \in V(G)$ have $c > 0$ children in T, and assume we have computed $\mathcal{R}(v_i)$ for each child v_i of v. Then $\mathcal{R}(v)$ can be computed in time at most $2^{\mathcal{O}(k^3)} \cdot c$.*

Proof sketch. Without loss of generality, let the open children of $v \in V(G)$ be $v_1, \ldots, v_t$, then $t \leq 2k$ by Point 3 of Observation 1. Let C be the set of remaining (closed) children of v, i.e. $C = \{v_{t+1}, \ldots, v_c\}$. Denote $V_i = V_{v_i}$, $\bar{V}_i = \bar{V}_{v_i}$, $i \in [c]$, and $V_0 = \delta(v) \setminus \cup_{i=1}^{c} V_i$. Note that all V_i, $i \in [c]_0$, are pairwise disjoint. If the record set of some child is empty, we conclude that $\mathcal{R}(v) = \emptyset$. Otherwise, we branch over all choices of $(R_i, S_i) \in \mathcal{R}(v_i)$ for each individual open child v_i of v. We also branch over all orientations R_0 of edges $\{uv | u \in V_0\}$.

Let $R' = \bigcup_{j\in[t]_0} R_j$. We branch over subsets S_0 of $V_0 \setminus \{v\}$ with precisely one incoming arc in R'. Let $S' =$

$\bigcup_{j\in[t]_0} S_j$, if R' is not anti-symmetric or contains two different arcs u_1w and u_2w for some $w \in S'$, we discard this branch. We also discard it if $V_i \cap S' \neq V_i \cap S_i$ for some $i \in [t]$. Otherwise, we create a trial record (R, S), where R is the restriction of R' to those arcs which have precisely one endpoint in V_v and $S = S' \cap \delta(v)$. We branch over 4 options for the in-neighbors of v in the partial solution, and in each case try to choose suitable records for the closed children.

Case 0: no in-neighbors. If v has no in-neighbors in R', and every closed child v_i contains the record $(\{vv_i\}, \{v_i\})$ (or $(\{vv_i\}, \emptyset)$ if $V_v(\{v_iv\}) = 0$), and every $w \in N_G(v) \setminus C$ with $\mathcal{V}_v(\{wv\}) > 0$ is in S', we add the record (R, S).

Case 1: one open in-neighbor. If there is precisely one incoming arc uv to v in R', and for every closed child v_i of v, $\mathcal{R}(v_i)$ contains the record $(\{vv_i\}, \emptyset)$, let $s_1 = \mathcal{V}_v(\{uv\})$. If $s_1 < \mathcal{V}_v(\{v_iv\})$ for some $v_i \in C$, and $\mathcal{R}(v_i)$ does not contain the record $(\{vv_i\}, \{v_i\})$, we discard this case. We also discard it if $s_1 < \mathcal{V}_v(\{wv\})$ for some $w \in N_G(v) \setminus (C \cup S')$. Otherwise, we add the records (R, S) and $(R, S \cup \{v\})$.

Case 1': one closed in-neighbor. If R' has no incoming arcs uv to v, we model partial solutions where the unique in-neighbor of v is one of its closed children. We branch over the choices of this closed child v_i. For every fixed $v_i \in C$ such that $\mathcal{R}(v_i)$ contains the record $(\{v_iv\}, \{v\})$, we compare $s_i = V_v(\{v_iv\})$ with all the values s_j over the rest of the closed children v_j of v. If $s_j > s_i$ for some j and $\mathcal{R}(v_j)$ does not contain the record $(\{vv_j\}, \{v_j\})$, we discard the case. We also discard it if for some $j \neq i$, $\mathcal{R}(v_j)$ does not contain $(\{vv_j\}, \emptyset)$. Otherwise, we compare s_i with $\mathcal{V}_v(\{wv\})$ for every $w \in N_G(v) \setminus C$. If for some $w \notin S'$ the latter is larger, we discard the branch as well. Otherwise, we add the record $(R, S \cup \{v\})$. If $\mathcal{R}(v_i)$ contains the record $(\{v_iv\}, \emptyset)$, we additionaly add (R, S).

Case 2: two or more in-neighbors. Finally, if $v \notin S'$, we model the subcase when v has more than one in-neighbor in a partial solution. For every closed child v_i of v such that $(\{v_iv\}, \emptyset) \notin \mathcal{R}(v_i)$, vv_i must be oriented towards v_i. On the other hand, for every $v_i \in C$ such that $\mathcal{R}(v_i)$ contains the record $(\{v_iv\}, \emptyset)$, by monotonicity of $\mathcal{V}_v$ we can just greedily orient the edge vv_i towards v. The corresponding value of v will be $s = \mathcal{V}_v(\{wv | wv \in R'\} \cup \{v_iv | v_i \in C, (\{v_iv\}, \emptyset) \in \mathcal{R}(v_i)\}$. If there is a closed child v_i such that $s < V_v(\{v_iv\})$ and $\mathcal{R}(v_i)$ does not contain the record $(\{vv_i\}, \{v_i\})$, we discard the branch. Moreover, if there is closed child v_i such that $(\{v_iv\}, \emptyset) \notin \mathcal{R}(v_i)$ and $(\{vv_i\}, \emptyset) \notin \mathcal{R}(v_i)$, we discard the branch. We also discard it if $s < V_v(\{vw\})$ for some $w \in N_G(v) \setminus C$ such that $w \notin S'$. Otherwise, we add the record (R, S).

For the running time, note that we branch over the choice of at most $2k + 1$ many binary relations R_i, and there are at most $\mathcal{O}(2^{(2k+2)^2})$ options for every such relation, which dominates the number of possible choices of subsets S_i of the boundaries. Therefore, we have at most $\mathcal{O}((2^{(2k+2)^2})^{2k+1} \leq 2^{\mathcal{O}(k^3)}$ branches. In each branch, we need at most linear in c time to traverse closed children of v. Hence, $\mathcal{R}(v)$ can be computed in time $2^{\mathcal{O}(k^3)} \cdot c$.

To establish the correctness of the procedure described above, which we will refer to as CRC(v) (combining records of children of v), we prove the following two claims:

Claim 1 ($\star$). *If $(R^*, S^*) \in \mathcal{R}(v)$, then CRC($v$) adds it.*

Claim 2 ($\star$). *If CRC(v) adds (R, S), then $(R, S) \in \mathcal{R}(v)$.*

This concludes the proof of Lemma 6. □

5 Discussion and Open Problems

The focus of this paper was the existence and complexity of EF1 and EFX orientations of goods, i.e., allocations that satisfy the corresponding fairness criterion and in addition, every agent gets items from their own predetermined set. In contrast to EFX orientations, which do not always exist and if they do it is almost always hard to find one, we have shown that EF1 orientations do exist and can be computed. Hence, EF1 orientations, in addition to fairness constraints, preserve some economic efficiency as well; however in Christodoulou *et al.* [2023] it was shown that EFX does not have the latter property, as in specific cases it *has* to give goods to agents that do not value them at all.

We conclude by highlighting some intriguing open problems that deserve further investigation.

From a purely theoretical point of view, the complexity of finding an EF1 orientation is open. Can we compute such an orientation in polynomial time? We currently do not know the answer even for three agents- for two agents the problem is easy. Alternatively, is the problem hard for some complexity class? Our proof indicates that the problem belongs to PLS, however, showing hardness is an intriguing question.

What about EF1 allocations that satisfy other types of constraints? Our results show existence under "approval" constraints for the agents. A different option would be to consider *cardinality* constraints for the agents, i.e., every agent should get a specific number of items. A version of this model was studied by Caragiannis and Narang [2024] and the existence of EF1 allocations is an open problem.

Moreover, in this paper we show weak NP-hard for EFX orientations where we have a small vertex cover. This leaves open whether we can compute if an EFX orientation exists when the weights are given in unary, i.e. can we find a pseudo-polynomial time algorithm? (However do note that, Caragiannis and Narang [2024] show that this is strongly NP-hard even for instances with constant max degree.)

The definition of EFX that both Christodoulou *et al.* [2023] and this paper has adopted, assumes that the envy should be eliminated by removing any item from an envied bundle. However, someone can study the relaxed notion of EFX+; recall in EFX+ the envy should be eliminated by removing any item that changes the value of the envied bundle. Our preliminary investigation shows that indeed this is a promising direction. We have identified some classes of (graph-based) monotone valuations where a natural generalization of EFX+ always exists and can be computed efficiently. This raises the natural question about for which classes of valuation functions EFX+ allocations are guaranteed to exist.

Acknowledgements

Argyrios Deligkas acknowledges the support of the EPSRC grant EP/X039862/1.

References

[Akrami *et al.*, 2022] Hannaneh Akrami, Rojin Rezvan, and Masoud Seddighin. An EF2X allocation protocol for restricted additive valuations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI'22*, pages 17–23. International Joint Conferences on Artificial Intelligence Organization, 2022.

[Akrami *et al.*, 2024] Hannaneh Akrami, Noga Alon, Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, and Ruta Mehta. Efx: a simpler approach and an (almost) optimal guarantee via rainbow cycle number. *Operations Research*, 2024.

[Amanatidis *et al.*, 2023] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence*, 322:103965, 2023.

[Bouveret and Lang, 2008] Sylvain Bouveret and Jérôme Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *Journal of Artificial Intelligence Research*, 32:525–564, 2008.

[Bouveret *et al.*, 2017] S Bouveret, K Cechlárová, E Elkind, A Igarashi, and D Peters. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI '17*, pages 135–141, 2017.

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

[Caragiannis and Narang, 2024] Ioannis Caragiannis and Shivika Narang. Repeatedly matching items to agents fairly and efficiently. *Theoretical Computer Science*, 981:114246, 2024.

[Caragiannis *et al.*, 2019a] Ioannis Caragiannis, Nick Gravin, and Xin Huang. Envy-freeness up to any item with high nash welfare: The virtue of donating items. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 527–545, 2019.

[Caragiannis *et al.*, 2019b] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation*, 7(3):1–32, 2019.

[Chaudhury *et al.*, 2021] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. *SIAM Journal on Computing*, 50(4):1336–1358, 2021.

[Chaudhury *et al.*, 2024] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. EFX exists for three agents. *Journal of the ACM*, 71(1):1–27, 2024.

[Christodoulou *et al.*, 2023] George Christodoulou, Amos Fiat, Elias Koutsoupias, and Alkmini Sgouritsa. Fair allocation in graphs. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pages 473–488, 2023.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, Cham, 2015.

[Deligkas *et al.*, 2021] Argyrios Deligkas, Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. The parameterized complexity of connected fair division. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI '21*, pages 139–145, 2021.

[Ganian and Korchemna, 2021] Robert Ganian and Viktoriia Korchemna. The complexity of bayesian network learning: Revisiting the superstructure. In *Proceedings of the 35th Conference on Neural Information Processing Systems, NeurIPS '21*, pages 430–442, 2021.

[Ganian and Korchemna, 2024] Robert Ganian and Viktoriia Korchemna. Slim tree-cut width. *Algorithmica*, pages 1–25, 2024.

[Goldberg *et al.*, 2023] Paul W Goldberg, Kasper Høgh, and Alexandros Hollender. The frontier of intractability for EFX with two agents. In *International Symposium on Algorithmic Game Theory*, pages 290–307. Springer, 2023.

[Johnson *et al.*, 1988] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

[Kyropoulou *et al.*, 2020] Maria Kyropoulou, Warut Suksompong, and Alexandros A Voudouris. Almost envy-freeness in group resource allocation. *Theoretical Computer Science*, 841:110–123, 2020.

[Lipton *et al.*, 2004] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In Jack S. Breese, Joan Feigenbaum, and Margo I. Seltzer, editors, *Proceedings of the 5th ACM Conference on Electronic Commerce, EC '04*, pages 125–131. ACM, 2004.

[Payan *et al.*, 2023] Justin Payan, Rik Sengupta, and Vignesh Viswanathan. Relaxations of envy-freeness over graphs. *AAMAS 2023*, pages 2652–2654, 2023.

[Plaut and Roughgarden, 2020] Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM J. Discret. Math.*, 34(2):1039–1068, 2020.

[Zhou *et al.*, 2024] Yu Zhou, Tianze Wei, Minming Li, and Bo Li. A complete landscape of EFX allocations of mixed manna on graphs. *arXiv preprint arXiv:2409.03594*, 2024.