Fair Division in a Variable Setting

Harish Chandramouleeswaran
Chennai Mathematical Institute
Chennai, India
harishc@cmi.ac.in

Prajakta Nimbhorkar*
Chennai Mathematical Institute
Chennai, India
prajakta@cmi.ac.in

Nidhi Rathi†
Max Planck Institute for Informatics, SIC
Saarbrücken, Germany
nrathi@mpi-inf.mpg.de

ABSTRACT

We study the classic problem of fairly dividing a set of indivisible items among a set of agents and consider the popular fairness notion of *envy-freeness up to one item* (EF1). While in reality, the set of agents and items may vary, previous works have studied *static* settings, where no change can occur in the system. We initiate and develop a formal model to understand fair division under the *variable input* setting: here, there is an EF1 allocation that gets disrupted because of the loss/deletion of an item, or the arrival of a new agent, resulting in a *near*-EF1 allocation. The objective is to perform a sequence of transfers of items between agents to regain EF1 fairness by traversing only via near-EF1 allocations. We refer to this as the EF1-Restoration problem.

In this work, we present algorithms for the above problem when agents have *identical monotone* valuations, and items are either all goods or all chores. Both of these algorithms achieve an optimal number of transfers (at most m/n, where m and n are the number of items and agents respectively) for identical additive valuations. Next, we consider a valuation class with graphical structure, introduced by Christodoulou et al. (EC 2023), where each item is valued by at most two agents, and hence can be seen as an edge between these two agents in a graph. Here, we consider EF1 *orientations* on (multi)graphs - allocations in which each item is allocated to an agent who values it. While considering EF1 *orientations on multigraphs* with additive binary valuations, we present an optimal algorithm for the EF1-Restoration problem. Finally, for monotone binary valuations, we show that the problem of deciding whether EF1-Restoration is possible is **PSPACE**-complete.

CCS CONCEPTS

• **Theory of computation → Algorithmic game theory**.

KEYWORDS

Fair division; EF1; Variable Input; Orientation; PSPACE

*Part of this work was carried out when PN was visiting the Max Planck Institute for Informatics, SIC, Saarbrücken, Germany.
†NR is supported by the Lise Meitner Postdoctoral Fellowship awarded by the Max Planck Institute for Informatics.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

ACM Reference Format:
Harish Chandramouleeswaran, Prajakta Nimbhorkar, and Nidhi Rathi. 2025. Fair Division in a Variable Setting. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Fair Division studies the fundamental problem of dividing a set of indivisible resources among a set of interested parties (often dubbed as *agents*) in a *fair* manner. The need of fairness is inherent in the design of many social institutions and occur naturally in many real-life scenarios like inheritance settlement, allocating radio and satellite spectrum, and air traffic management, to name a few [13, 19, 29, 30, 38]. While the first mentions of fair division date back to the Bible and Greek mythology, its first formal study is credited to Steinhaus, Banach, and Knaster in 1948 [35]. Since then, the theory of fair division has enjoyed flourishing research from mathematicians, social scientists, economists, and computer scientists alike. The last group has brought a new flavor of questions related to different models of computing *fair* allocations. We refer to [3, 10, 11, 31] for excellent expositions and surveys of fair division.

Envy-freeness is one of the quintessential notions of fairness for resource allocation problems that entails every agent to be *happy* with their own share and not prefer any other agent's share over theirs in the allocation, i.e., being *envy-free* [20]. This notion has strong existential properties when the resource to be allocated is divisible, like a *cake* [36, 37]. But, a simple example of two agents and one item, shows that envy-free allocations may not exist for the case of indivisible items. Hence, several variants of envy-freeness have been explored in the literature. *Envy-freeness up to some item* (EF1) is one such popular relaxation [12] that entails an allocation to be *fair* when any envy for an agent must go away after removing some item from the envied bundle. When we have an EF1 allocation, we say that the agents are EF1-happy, while if an agent's envy does not go away after removing some item from the envied bundle, we refer to it as EF1-*envy*. It is known that EF1 allocations always exist and can be computed efficiently as well [27].

Given that the notion of EF1 is tractable, it becomes a tempting problem to explore it in a *changing* environment - that is, what happens when we have a *variable* input of items and agents, as is natural in many real-world scenarios. This work aims to develop tools for the problem of *restoring fairness* in the system when changes occur, with *minimal disruption to the existing allocation.*

We initiate the study of fair division of indivisible items in a changing environment due to variable input. In this work, we go beyond the classic *static* setting of fair division and study an interesting variant where an agent may lose an item in a given EF1 allocation, and therefore, may no longer be EF1-happy. Another such scenario is when a new agent enters the system with an empty bundle to begin with, and starts EF1-envying some of the existing agents. We call such allocations *near*-EF1 allocations - where every agent but one (say, the new agent, or the agent who lost an item), is EF1-happy.

Given a near-EF1 allocation, the goal is to redistribute the items by causing *minimal disruption* to the existing allocation. In particular, the redistribution needs to be carried out while ensuring the EF1-happiness of other agents (i.e., creating no new EF1-envy), and to finally reach an EF1 allocation. Starting from a near-EF1 allocation where every agent except one, say agent 1, is EF1-happy, we allow a sequence of *transfers* of a single item from one bundle to another, such that the new allocation is near-EF1, i.e., every agent other than agent 1 remains EF1-happy. Our aim is to modify the allocations by performing a sequence of transfers between agents so as to eventually reach an EF1 allocation.

Note that it is always possible to start from scratch and find an EF1 allocation (say, by using the algorithm in [27]), but this leads to an excessive and undesirable (and possibly unnecessary) amount of change in the bundles of all the agents. Hence, restoring fairness after any change in the input by ensuring minimal modifications to what is already allocated is desirable. This is well-motivated from a practical perspective, both in the case of goods as well as chores. Consider a company where various people work on different tasks simultaneously. When a task is completed, the person who was working on it has less work at hand, and hence, the task allocation may no longer be fair. It is imperative to redistribute the tasks with as little reshuffling as possible so as to restore fairness. Similarly, when goods get damaged, stolen, or expire, the fairness guarantees must be restored with as little reshuffling of goods as possible. For instance, consider the scenario in which people are allocated slots for their computing tasks on various machines. If a machine stops working, then it is desirable to redistribute the slots on the working machines so as to regain fairness, and also minimize the disturbance caused to others.

To avoid confusion, we remark that a different non-static setting of *online* fair division has also been studied, where items or agents arrive and depart over time and the aim is to dynamically build (from scratch) a fair division against an uncertain future; see the survey [1]. Another line of research is to explore fair division protocols that remain consistent with *population monotonicity* and *resource monotonicity*, where the set of agents or items may vary [15].

Our Contribution: In this work, we develop a formal model of fair division under variable input, where an agent loses an item, or a new agent with an empty bundle enters the system. This leads to a *near*-EF1 allocation, where all agents but one are EF1-happy. The goal is to restore EF1 fairness guarantees via a sequence of transfers of items while maintaining near-EF1 allocations - we call this the EF1-Restoration problem. If this is possible for a given instance, we say EF1-Restoration admits a positive solution. Our main results are given below. Proofs omitted due to space constraints appear in the full version [16].

(1) For the EF1-Restoration problem with identical monotone valuations, we present an algorithm (Algorithm 1) to restore EF1. Moreover, for identical additive valuations, our algorithm is optimal with m/n transfers. This remains true when all the items are *chores* ([16], Appendix A). On the other hand, a positive result doesn't hold in the *mixed setting* with both goods and chores, or for the analogous EFX-Restoration problem for goods, even in the case of identical additive valuations ([16], Appendix B).
(2) We consider a class of *graphical* valuations introduced by Christodoulou et al. [17], where every item is positively valued by at most two agents, and hence can be represented as an edge between them. Moreover, let the initial near-EF1 allocation have the property that each item is allocated to an agent who values it positively (known as an *orientation* [18, 40]). In this setting, we present an optimal algorithm (Algorithm 2) to restore EF1 with $n-1$ transfers (when agent 1 loses one item), while maintaining a near-EF1 *orientation* at each step.
(3) We prove that deciding if EF1-Restoration is possible is **PSPACE**-complete for *monotone binary* valuations (Section 6).

Related Work: The closest work to ours is by Igarashi et al. [24], in which they consider the problem of deciding whether one EF1 allocation can be reached from another EF1 allocation via a sequence of exchanges such that every intermediate allocation is EF1. They prove reachability is guaranteed for two agents with identical or binary valuations, as well as for any number of agents with identical binary valuations. Ito et al. [25] also consider the problem of reachability via exchanges, with the allocations satisfying some different constraints. We ask a similar question in a variable input setting, in which we start with a near-EF1 allocation, and the goal is to reach an EF1 allocation via a sequence of near-EF1 allocations.

As mentioned earlier, several non-static models of fair division have been studied. One such model considers the online nature of either items or agents arriving or departing over time, in which one needs to dynamically allocate the items in a fair manner against an uncertain future. There are various works that tackle different dimensions of the problem: (i) resources being divisible [39] or indivisible [2, 26], (ii) resources being fixed and agents arriving over time [39], or agents being fixed and resources arriving over time [1], or both arriving online [28], and (iii) whether mechanisms are informed [39] or uninformed [23] about the future.

Another line of work studies resource- and population-monotonicity, in which the set of agents or items may change. The goal is construct fair division protocols in which the utility of all participants change in the same direction - either all of them are better off (if there is more to share, or fewer to share among), or all are worse off (if there is less to share, or more to share among); see [15, 33, 34].

The notion of fairness considered in our work as well as in the results mentioned above is envy-freeness up to one item (EF1). Since its introduction by Budish [12], it has been extensively studied in the literature in various settings ([14], [6], [22], to mention a few). For goods, a polynomial-time algorithm to compute an EF1 allocation was given by Lipton et al. [27]. For a mixture of goods and chores, Aziz et al. [5] defined an analogous notion of EF1 and gave a polynomial-time algorithm to compute an EF1 allocation in the case of additive valuations. For chores (resp. a mixture of goods and chores), a polynomial-time algorithm to compute EF1 allocations for monotone (resp. doubly monotone) valuations was given by Bhaskar et al. [7].

As mentioned earlier, the class of *graphical* valuations, in which every item is valued by at most two agents and hence can be seen as an edge between them, was introduced in [17] in the context of *envy-freeness up to any item* (EFX) allocations. They proved the existence of EFX allocations on simple graphs. This work has motivated studies on EFX and EF1 orientations and allocations on graph settings - e.g. Zhou et al. [41] studied the mixed manna setting with both goods and chores and proved that determining the existence of EFX orientations for agents with additive valuations is **NP**-complete and provide certain tractable special cases. Zeng et al. [40] relate the existence of EFX orientations and the chromatic number of the graph. Recently, Deligkas et al. [18] showed that EF1 orientations always exist for monotone valuations and can be computed in pseudopolynomial time. We refer the readers to the excellent surveys by Biswas et al. [8] on fair division under such "structured set constraints", and by Amanatidis et al. [3] and Guo et al. [21] on various notions of fairness in the settings of both goods and chores.

2 NOTATION AND PRELIMINARIES

For a positive integer k, we write $[k]$ to denote the set $\{1, 2, \dots, k\}$, and we write 2^S to denote the power set of a set S.

Consider a set $\mathcal{G}$ of m *indivisible* items that needs to be assigned to a set $[n]$ of n agents fairly. The preferences of an agent $i \in [n]$ over these items is specified by a valuation function $v_i\colon 2^{\mathcal{G}} \to \mathbb{R}^+ \cup \{0\}$, i.e., $v_i(S)$ denotes the value agent $i \in [n]$ associates to the set of items $S \subseteq \mathcal{G}$. We define an *allocation* $X = (X_1, \dots, X_n)$ to be a partition of the set $\mathcal{G}$ among the n agents, where X_i corresponds to the bundle assigned to agent $i \in [n]$. The goal is to construct "fair", i.e., *envy-freeness up to one item* (EF1) allocations (see Definition 2.3). We call the tuple $\mathcal{I} = \left\langle [n], \mathcal{G}, \{v_i\}_{i\in[n]} \right\rangle$ a *fair division instance*.

For an item $g \in \mathcal{G}$, we denote $v_i(\{g\})$ simply by $v_i(g)$ for a cleaner presentation.

We say that a valuation function v is *monotone* if $v(S) \leq v(T)$ for all $S, T \subseteq \mathcal{G}$ with $S \subseteq T$. We say that v is *binary* if, for each $S \subseteq \mathcal{G}$, and for each $g \in \mathcal{G}$, we have $v(S \cup g) - v(S) \in \{0, 1\}$, i.e., if each item $g \in \mathcal{G}$ adds a marginal value of either 0 or 1 to any bundle. A natural subclass of monotone valuations is the class of *additive* valuations - we say v is *additive* if, for all $S \subseteq \mathcal{G}$, we have $v(S) = \sum_{g\in S} v(g)$. Observe that an additive valuation is binary if and only if $v(g) \in \{0, 1\}$, for each $g \in \mathcal{G}$.

We study the class of *graphical valuations* [17] in this work, that we now define. These are inspired by geographic contexts, where agents only care about resources nearby and show no interest in those located farther away.

Definition 2.1 (Graphical Valuation)**.** A graphical valuation is one that is representable as a (multi)graph where vertices correspond to the agents, and edges correspond to the items. An item is valued positively only by the two agents incident to the corresponding edge. In other words, for a multigraph $G = (V, E)$, each vertex $i \in V$ has a valuation v_i such that $v_i(g) > 0$ if and only if g is incident to i in G.

Definition 2.2 (Envy and EF1-envy)**.** Agent $i \in [n]$ is said to *envy* agent $j \in [n]$ in an allocation X if i values j's bundle strictly more than their own bundle, i.e., $v_i(X_i) < v_i(X_j)$. Furthermore, we say i *EF1-envies* j if, for all $g \in X_j$, we have $v_i(X_i) < v_i(X_j \setminus g)$, i.e., i continues to envy j even after virtually removing any one item from j's bundle. We say i is *EF1-happy* in X if they don't EF1-envy any other agent.

We now define the fairness notion of *envy-freeness up to one item* (EF1).

Definition 2.3 (Envy-freeness up to one item (EF1) [12])**.** An allocation X is said to be EF1 if every agent is EF1-happy in X.

A special type of allocation called an *orientation* has been studied recently, and was introduced by Christodoulou et al. [17] while studying EFX allocations in the context of graphical valuations. There has been further work on EFX orientations by Zeng and Mehta [40], and was first studied in the EF1 setting by Deligkas et al. [18].

Definition 2.4 (Orientation)**.** An allocation X is an *orientation* if every item is allocated to an agent who has positive marginal value for it.

An orientation is a type of *non-wasteful allocation*. Note that orientations for graphical valuations with graph G, are equivalent to assigning a direction to each edge in G, such that the edge is directed towards the agent who owns the item in a particular allocation.

We next define the concept of *envy graph* of an allocation.

Definition 2.5. For an allocation $X = (X_1, \dots, X_n)$, its *envy graph* $\mathrm{G}_{\text{envy}}(X)$ is a directed graph with agents appearing as vertices. We add a directed edge from agent i to agent j if i envies j, i.e., $v_i(X_i) < v_i(X_j)$.

3 OUR PROBLEM AND CONTRIBUTION

In this section, we define the problem and state our results formally. Recall that our variable input setting involves an initial EF1 allocation, from which, either an item allocated to an agent is lost, or a new agent with an empty bundle arrives. Without loss of generality, let this be agent 1.

We will now introduce our notion of a "near-EF1" allocation, which is the primary object of study in this work.

Definition 3.1 (Near-EF1 allocation)**.** An allocation X is said to be *near*-EF1 if every agent is EF1-happy in X except possibly one *fixed* agent, say agent 1, who we refer to as the *unhappy agent*.

Definition 3.2 (Amount of EF1-envy)**.** Let $X = (X_1, \dots, X_n)$ be an allocation. The amount of EF1-envy that agent i has towards agent j in X, denoted by $\varepsilon_{i\to j}(X)$, is defined as
$\varepsilon_{i\to j}(X) := \min_{g\in X_j} \left\{ v_i(X_j \setminus g) - v_i(X_i) \right\}$.

Note that X is an EF1 allocation, if and only if $\varepsilon_{i\to j}(X) \leq 0$ for each $i, j \in [n]$.

3.1 Problem Setup

Given a near-EF1 allocation, the goal of this work is to restore EF1 by redistributing the items in a way that causes *minimal disruption* to the system, that we formally define next.

Definition 3.3 (Valid Transfer). Let X be a near-EF1 allocation with one unhappy agent - agent 1. Then, the transfer of an item from one agent to another is said to be a *valid transfer*, if the resulting allocation Y is a near-EF1 allocation, and the amount of EF1-envy agent 1 has towards other agents is no larger in Y as compared to that in X.

Let us now formally conceptualize the problem of fair division in a variable setting that forms the focus of this work.

Problem 3.4 (The EF1-Restoration problem). Given a fair division instance, and a near-EF1 allocation X, where every agent except agent 1 is EF1-happy, determine if it is possible to reach an EF1 allocation by a sequence of valid transfers.

3.2 Technical Contribution

We begin by studying identical valuations and prove that any EF1-Restoration instance has a positive solution here. Formally, we prove the following (in Section 4).

Theorem 3.5. *Given a fair division instance with an identical monotone valuation v, and any near-*EF1 *allocation X, the* EF1-Restoration *problem always admits a positive solution. Moreover, it is possible to ensure that the unhappy agent* 1 *is the only recipient in each of these valid transfers, and the transfer is always made from an agent i whom agent* 1 EF1*-envies (i.e., $\varepsilon_{1\to i} > 0$).*

Furthermore, if v is additive, at most m/n valid transfers to agent 1 *suffice to reach an* EF1 *allocation. This is tight in the worst case.*

The above theorem also holds for the case in which all the items are *chores*. The proof is analogous ([16], Appendix A). However, a positive result doesn't hold in the *mixed setting* with both goods and chores, or for the analogous EFX-Restoration problem for goods, even in the case of identical additive valuations ([16], Appendix B).

Next (in Section 5), we identify another interesting valuation class where we can achieve EF1-Restoration. We study EF1 orientations in graphical setting, with additive binary valuations and prove the following.

Theorem 3.6. *Consider a fair division instance on multigraphs with additive binary valuations. Given any near-*EF1 *orientation X, the* EF1-Restoration *problem always admits a positive solution. And, an* EF1*-orientation may be restored by effecting at most $K(n-1)$ valid transfers, where $K = \max_{j\in[n]} \varepsilon_{1\to j}$ in X. Furthermore, this bound is tight and each transfer results in a near-*EF1 *orientation.*

In Section 6, we prove that EF1-Restoration is hard in general, by considering the subclass of binary monotone valuations, in which every subset of the set of items is valued at either 0 or 1. It turns out that the EF1-Restoration problem is **PSPACE**-complete even in this case.

Theorem 3.7. *The* EF1-Restoration *problem is* **PSPACE***-complete for monotone binary valuations.*

In fact, the result holds even if *valid exchanges*, apart from valid transfers are allowed. We present a reduction from the **PSPACE**-complete problem of Perfect Matching Reconfiguration [9]. We have an involved construction of the valuation functions in our EF1-Restoration instance, so that we prohibit all problematic transfers/exchanges, and ensure that *every* EF1 allocation that may be constructed from our instance corresponds to *exactly one* perfect matching - the target perfect matching in the Perfect Matching Reconfiguration instance with which we begin.

4 EF1-RESTORATION FOR IDENTICAL VALUATIONS

In this section, we first consider identical monotone valuations and give a constructive proof (via Algorithm 1) for the EF1-Restoration problem (in Lemma 4.1), and then bound the number of transfers required in the additive case (in Lemma 4.2). These two lemmas together establish Theorem 3.5.

Let us denote the given near-EF1 allocation by X wherein agent 1 is unhappy. In each step of Algorithm 1, an agent $i \neq 1$ is chosen who would be *least affected* by giving away their most valuable item, denoted $g^X_{i,\text{best}}$, to agent 1. Formally, let $g^X_{i,\text{best}} := \arg\min_{g\in X_i} \{v(X_i \setminus g)\}$. The item $g^X_{i,\text{best}}$ is then transferred to agent 1. The crux of our proof is to show that the above transfer is valid.

Algorithm 1: EF1-Restoration for identical valuations

Input : A fair division instance $\mathcal{I} = \langle [n], \mathcal{G}, v\rangle$ with identical monotone (positive) valuation v and a near-EF1 allocation $X = (X_1, \ldots, X_n)$ with agent 1 being unhappy.

Output: An EF1 allocation in $\mathcal{I}$.

```
1 EF1_Restorer(X):
2     S ← agents that agent 1 EF1-envies in X
3     while S ≠ ∅ do
4         i ← argmax_{k∈[n]\{1}} {v(X_k \ g^X_{k,best})}
5         X_i ← X_i \ g^X_{i,best}
          X_1 ← X_1 ∪ g^X_{i,best}     } Transfer g^X_{i,best} from i to 1
6         Update S
7     end
8     return X = (X_1, ..., X_n)
```

Lemma 4.1. *The* EF1-Restoration *problem admits a positive solution for identical monotone (positive) valuations.*

Proof. Let $X = (X_1, \ldots, X_n)$ be the input near-EF1 allocation, with a positive valuation v that is common to all the agents. Let $i := \arg\max_{k\in[n]\setminus\{1\}} \left\{v(X_k \setminus g^X_{k,\text{best}})\right\}$. That is, in the allocation X, agent i is the one (other than agent 1, of course) who would possess the most valuable bundle, even after losing their most valuable item. As indicated in Line 5 of Algorithm 1, we transfer $g^X_{i,\text{best}}$ from i to agent 1, and denote the resulting allocation by Y. We will prove that this is a valid transfer, i.e., Y is near-EF1 and the transfer does not lead to an increase in the amount of EF1-envy agent 1 has towards any other agent.

- We first show that agent i does not EF1-envy any agent $j \neq 1$ after the transfer. We have $Y_1 = X_1 \cup g^X_{i,\text{best}}$, $Y_i = X_i \setminus g^X_{i,\text{best}}$, and $Y_j = X_j$ for all $j \in [n] \setminus \{1, i\}$. Then, we have, for each $j \notin \{1, i\}$,

$$v(Y_i) = v(X_i \setminus g^X_{i,\text{best}}) \geq v(X_j \setminus g^X_{j,\text{best}}) = v(Y_j \setminus g^Y_{j,\text{best}}), \quad (1)$$

where the inequality in (1) follows from the choice of i in Line 4 of Algorithm 1. Hence, agent i does not EF1-envy any agent $j \neq 1$ after the transfer.

- Next, we prove that no agent EF1-envies agent 1 after the transfer (i.e., in Y). Note that the transfer was made because agent 1 had EF1-envy towards some agent, say agent 2 in X. So, for each $j \notin \{1, i\}$, we have the following:

$$\begin{aligned} v(Y_j) = v(X_j) \geq v(Y_i) &= v(X_i \setminus g^X_{i,\text{best}}) && (X \text{ is near-EF1}) \\ &\geq v(X_2 \setminus g^X_{2,\text{best}}) && (\text{by our choice of } i) \\ &> v(X_1) && (\text{agent 1 EF1-envies agent 2}) \end{aligned}$$

and

$$v(X_1) = v(Y_1 \setminus g^X_{i,\text{best}}) \geq v(Y_1 \setminus g^Y_{1,\text{best}}) \qquad (X_1 \subset Y_1)$$

Therefore, we have $v(Y_j) \geq v(Y_1 \setminus g^Y_{1,\text{best}})$ for all $j \neq 1$.

- Observe that agent 1 must have had EF1-envy towards agent i. As assumed earlier, let agent 2 be one of the agents who is EF1-envied by agent 1. So, $v(X_2 \setminus g^X_{2,\text{best}}) > v(X_1)$. But, since agent i was chosen for the transfer, we must have had $v(X_i \setminus g^X_{i,\text{best}}) \geq v(X_2 \setminus g^X_{2,\text{best}}) > v(X_1)$.
- Since 1 has gained an item, the EF1-envy agent 1 has towards any other agent in Y cannot be larger than that in X.

Overall, we have shown that the transfer in Line 5 is valid and results in an increase of one item in the bundle of agent 1 (as long as agent 1 is not EF1-happy). Hence, Algorithm 1 eventually terminates with an EF1 allocation. ◻

Next, we give an upper bound on the number of transfers made by Algorithm 1 in the case of identical additive valuations in Lemma 4.2, and prove that this is tight.

Lemma 4.2. *Algorithm 1 reaches an* EF1 *allocation by making at most m/n valid transfers to agent* 1 *for the* EF1-Restoration *problem with identical additive valuations. This bound is tight.*

Proof. Let X be the input near-EF1 allocation, and Y be the output (EF1) allocation of Algorithm 1. We write v to denote common additive valuations of the agents. To bound the number of transfers executed in the worst-case, we may assume (i) $X_1 = \emptyset$, since otherwise, we may only have to make fewer transfers and (ii) all items in $\mathcal{G}$ are valued the same. This means that, at each iteration, the most valuable item that is transferred from a bundle, is no more valuable than the least valuable item in all of $\mathcal{G}$, thereby requiring more transfers to 1 to restore EF1. Given that we assume (ii), we may further assume that $v(g) = 1$ for every $g \in \mathcal{G}$.

Let $i \in [n]$. Let $k_i \coloneqq v(X_i) = |X_i|$, and let $k'_i \coloneqq v(Y_i) = |Y_i|$. By (i), we have $k_1 = 0$, and hence the number of transfers made by Algorithm 1 is k'_1 (recall that every transfer is to agent 1). Since Y is EF1, and each item is valued at 1 by all agents, it follows that $k'_i = k'_1$ or $k'_i = k'_1 + 1$. Hence,

$$nk'_1 \leq \sum_{i=1}^{n} k'_i = m,$$

so that $k'_1 \leq \frac{m}{n}$, which is what we set out to prove.[1]

This bound is tight (see Footnote 1). Consider a near-EF1 allocation X in which $X_1 \coloneqq \emptyset$, and each of $X_2, \ldots, X_n$ contain m/n items, each of same value. Here, the transfer of $\lceil \frac{m-n+1}{n} \rceil$ goods to the unhappy agent 1 is necessary to restore EF1. ◻

Remark 4.3. Theorem 3.5 also holds in the case of *chores* (i.e., when v is an identical monotone (*negative*) valuation) on similar arguments, and hence deferred to the full version of this work (see [16], Appendix A).

Remark 4.4. Even with identical additive valuations, (i) EF1-Restoration in the *mixed* setting (with both *goods and chores*), and (ii) the analogous EFX-Restoration problem for goods may not admit a positive solution. This discussion is deferred to the full version ([16], Appendix B).

5 EF1-RESTORATION FOR ORIENTATIONS UNDER GRAPHICAL VALUATIONS

In this section, we consider the valuation class defined using multigraphs where an item is valued by at most two agents and consider additive binary valuations. Here, any item g corresponds to an edge (i, j) between two agents i and j with $v_i(g) = v_j(g) = 1$, while $v_k(g) = 0$ for all agents $k \neq i, j$.

We study a special class of non-wasteful allocations, called orientations. We prove that the EF1-Restoration problem admits a positive solution for this setting (Theorem 3.6). We do so by developing Algorithm 2, that achieves an optimal number of transfers (in the worst case) to reach an EF1 allocation. Note that EF1 orientations always exist for monotone valuations, as proved recently in [18].

Before describing our algorithm, we prove the following properties of orientations for additive binary valuations.

Lemma 5.1. *For fair division instances with graphical structure, let v_i be the additive binary valuation of an agent $i \in [n]$, and let X be an orientation. Then, we have*

1. *For any $i, j \in [n]$, $v_i(X_i) = |X_i| \geq v_j(X_i)$. That is, if the allocation is an orientation, every agent values their bundle at least as much as any other agent values it.*
2. *If there exists a (directed) path $(i_0, i_1, i_2, \ldots, i_d)$ of length d from agent i_0 to agent i_d in the envy graph $\mathrm{G}_{envy}(X)$, then $v_{i_d}(X_{i_d}) \geq v_{i_0}(X_{i_0}) + d$.*
3. *The envy graph $\mathrm{G}_{envy}(X)$ is acyclic.*

Proof. (1) Since X is an orientation, and the valuation is binary, we have $v_i(X_i) = |X_i|$. Note that, for any other agent $j \neq i$, we have $v_j(g) \in \{0, 1\}$ for any $g \in X_i$. Hence, we have $v_j(X_i) \leq |X_i| = v_i(X_i)$.

(2) Assume agent i envies agent j. Then, using part (1), we have $|X_i| = v_i(X_i) < v_i(X_j) \leq v_j(X_j) = |X_j|$. Therefore, we obtain $v_j(X_j) \geq v_i(X_i) + 1$. Generalizing it to a d-length envy path in $\mathrm{G}_{\text{envy}}(X)$, say $(i_0, i_1, \ldots, i_d)$, we get the following:

$$v_{i_d}(X_{i_d}) \geq v_{i_{d-1}}(X_{i_{d-1}}) + 1 \geq v_{i_{d-2}}(X_{i_{d-2}}) + 2 \ldots \geq v_{i_0}(X_{i_0}) + d$$

[1] In fact, the EF1 allocation with $k'_2 = \ldots = k'_n = k'_1 + 1$ will be reached by Algorithm 1 before the EF1 allocation having n identical bundles. Solving $k'_1 + (n-1)(k'_1 + 1) \leq m$ would then give us $k'_1 \leq \frac{m-n+1}{n}$, which is the exact bound. The given example instance shows that this is tight.

(3) Let us now assume a cycle $(i_0, \dots, i_d)$ in $\mathrm{G}_{\text{envy}}(X)$, where agent i_d envies i_0. Using part (2), we can first write $v_{i_d}(X_{i_d}) \geq v_{i_0}(X_{i_0}) + d$. And then using part (1), we have $v_{i_0}(X_{i_0}) \geq v_{i_d}(X_{i_0})$. Combining the last two inequalities, we obtain $v_{i_d}(X_{i_d}) \geq v_{i_d}(X_{i_0})$. This means i_d does not envy i_0, leading to a contradiction. Hence, G_{envy} is acyclic. □

Algorithm 2: EF1-Restoration on multigraphs

Input : A fair division instance $\mathcal{I} = \langle [n], \mathcal{G}, \{v_i\}_i \rangle$ on a multigraph with additive binary valuations and a near-EF1 orientation $X = (X_1, \dots, X_n)$.
Output : An EF1-orientation.

EF1_Orientation_Restorer(X):
  $S \leftarrow$ agents EF1-envied by 1 in X
  while $S \neq \emptyset$ **do**
    $s \leftarrow$ closest sink to 1 in $\mathrm{G}_{\text{envy}}(X)$
    $\mathcal{P} \leftarrow$ shortest path from 1 to s in $\mathrm{G}_{\text{envy}}(X)$
    $p \leftarrow$ the predecessor of s on $\mathcal{P}$
    $g \leftarrow$ an item in X_s with $v_p(g) = 1$
    $\left.\begin{array}{l} X_s \leftarrow X_s \setminus g \\ X_p \leftarrow X_p \cup g \end{array}\right\}$ Transfer of g from s to p
    Update S
  end
  return $X = (X_1, \dots, X_n)$

With Lemma 5.1 at our disposal, we are ready to prove our main theorem. We present Algorithm 2 for the EF1-restoration problem and prove the following.

Theorem 5.2. *For the* EF1-Restoration *problem on a multigraph with additive binary valuations, Algorithm 2 terminates with an* EF1*-orientation, and it transfers at most $K(n-1)$ items, where $K := \max_{j \in [n]} \{\varepsilon_{1 \to j}\}$ in the given near-*EF1 *orientation X.*

Proof. Let X be the given near-EF1 orientation. Since $K = \max_{j \in [n]} \{\varepsilon_{1 \to j}\}$ in X, and each item is valued at most 1, we know that at least K transfers to agent 1 are necessary to restore EF1. We prove that it is sufficient as well.

Using Lemma 5.1, we know that $\mathrm{G}_{\text{envy}}(X)$ is acyclic. Note that the transfer of an item from an agent i who envies some j in X is not valid since it creates EF1-envy from i to j. So, a valid transfer can only be from a sink in $\mathrm{G}_{\text{envy}}(X)$. However, agent 1 may not positively value any item in the bundle of any sink. Therefore, in Algorithm 2, we transfer an item from a sink to its predecessor in $\mathrm{G}_{\text{envy}}(X)$. Since the predecessor must positively value at least one item from the sink's bundle, the orientation property is maintained along with the near-EF1 condition. We show that such transfers are always valid. Furthermore, we show that agent 1 will receive an item after at most $n-1$ (valid) transfers. Hence, after $K(n-1)$ transfers, we must reach an EF1 orientation.

Let $s \in [n]$ be a sink that is reachable from agent 1 in $\mathrm{G}_{\text{envy}}(X)$. We note that $s \neq 1$ as otherwise, X is already an EF1-orientation. Furthermore, among all sinks reachable from 1 in $\mathrm{G}_{\text{envy}}(X)$, let s be a closest sink to 1. Let $\mathcal{P}$ be a shortest path from 1 to s, say of length d. Let p be the predecessor of s on $\mathcal{P}$. Since p envies s, there exists an item $g \in X_s$ such that $v_p(g) = 1$. Moreover, since X is an orientation, we know $v_s(g) = 1$ as well. Now, since we have graphical valuations, every other agent $j \neq p, s$ must value g at zero. Hence, after this transfer, the resulting allocation (Y, say) remains an orientation.

We have $Y_s = X_s \setminus g$, $Y_p = X_p \cup g$, and $Y_j = X_j$ for all $j \neq p, s$. Since g is valued only by p and s, no agent other than s can have any EF1-envy towards p. Using Lemma 5.1, we know $v_s(X_s) \geq v_p(X_p) + 1$. Hence,

$$v_s(Y_s) = v_s(X_s) - 1 \geq v_p(X_p) = v_p(Y_p \setminus g)$$

That is, s has no EF1-envy towards p in Y. Therefore, the transfer of g from s to p is a valid transfer.

If $p \neq 1$, p does not envy anyone in Y. To see this, note that p has no EF1-envy towards anyone, including s in X. So, for any agent t who p envies in X, we have $v_p(X_p) \geq v_p(X_t) - 1$, and $v_p(Y_p) = v_p(X_p) + 1 \geq v_p(X_t) = v_p(Y_t)$. Therefore, p becomes a sink in $\mathrm{G}_{\text{envy}}(Y)$.

Observe that, in Y, p is a sink which has a path of length $d-1$ from 1 in $\mathrm{G}_{\text{envy}}(Y)$. Thus, after each valid transfer of an item from a closest sink to its predecessor, the distance between agent 1 to its closest sink in the envy graph decreases by 1. As soon as one of the successors of agent 1 becomes a sink, 1 can receive an item and become EF1-happy. Since the closest sink in $\mathrm{G}_{\text{envy}}(X)$ can be at distance at most $n-1$, clearly $n-1$ transfers from a closest sink to its predecessor suffice, to give one item to agent 1. Since the recipient always values the transferred item positively, the orientation property is maintained. Overall, K items can be transferred to agent 1 via $K(n-1)$ valid transfers and this results in an EF1 orientation. This completes our proof. □

5.1 Lower bound on the transfer complexity

In this section, we give an example to show that the bound on transfers given in Theorem 5.2 is tight. In particular, we prove the following result.

Theorem 5.3. *For every $n \geq 2$, there exists a near-*EF1 *orientation with $m = \frac{n^2+3n-2}{2}$, such that $n-1$ valid transfers are required for transferring one item to agent 1 that she positively values, so that each intermediate allocation is a near-*EF1 *orientation. This is true for additive binary valuations.*

Proof. Consider a fair division instance with n agents and a near-EF1 orientation X as follows. Let $|X_1| = 1$ and $|X_i| = i+1$ for all $i \neq 1$. For an agent $i \neq n$, we define her valuation v_i as,

$$v_i(g) = \begin{cases} 1, & \text{if } g \in X_i \cup X_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

Finally, for agent n, we define $v_n(g) = 1$ if and only if $g \in X_n$. It can be seen that $\mathrm{G}_{\text{envy}}(X)$ consists of a single envy path with 1 as its source and n as its sink. Also, X is a near-EF1 orientation since every agent other than 1 is EF1-happy. In particular, agent 1 has EF1-envy towards agent 2 with $\varepsilon_{1 \to j} = 1$.

To have every intermediate allocation be an orientation, an agent may receive any item only from her successor in the envy graph.

Thus, agent 1 may receive an item only from agent 2, since she does not value any item in any other agent's bundle. However, agent 2 may not transfer an item to agent 1 unless she receives an item from agent 3, and in general, any agent $i \neq n$ may not transfer an item to her predecessor, as this creates EF1-envy from i towards her successor in $G_{\text{envy}}(X)$. Therefore, the only way to restore an EF1 orientation is to transfer an item from agent n to agent $n-1$, followed by a transfer from agent $n-1$ to agent $n-2$ and so on, and finally from agent 2 to agent 1. These are a total of $n-1$ valid transfers, thereby completing the proof.

Below, we show this example with three agents and eight items.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
Agent 1	**1**	1	1	1	0	0	0	0
Agent 2	0	**1**	**1**	**1**	1	1	1	1
Agent 3	0	0	0	0	**1**	**1**	**1**	**1**

□

Theorem 5.2 and Theorem 5.3 together imply Theorem 3.6.

6 PSPACE-COMPLETENESS

In this section, we show that the EF1-Restoration problem may not always have an affirmative answer, and in fact, it is **PSPACE**-complete to decide so, even with monotone binary valuation functions, and even when *valid exchanges* of single items are permitted, in addition to valid transfers. Our reduction is inspired by Igarashi et al. [24], in which they give a polynomial-time reduction from the **PSPACE**-complete problem of Perfect Matching Reconfiguration [9] to the problem of deciding if there exists an EF1 exchange path between two fixed EF1 allocations, with general additive valuations.

Their problem involves deciding if there is a sequence of exchanges between two given EF1 allocations. We would like to prove that deciding if *any* EF1 allocation (as opposed to some fixed EF1 allocation, as in [24]) is reachable from a given near-EF1 allocation via transfers *and* exchanges (as opposed to reachability only via exchanges in [24]) is **PSPACE**-complete, provided we maintain a near-EF1 allocation after each operation. This leads to an involved construction of the valuation functions in our EF1-Restoration instance, so as to ensure that *every* EF1 allocation that may be constructed from our instance corresponds to *exactly one* perfect matching - the target perfect matching in the Perfect Matching Reconfiguration instance with which we begin.

Proof of Theorem 3.7. *Containment in* **PSPACE**: We first show that the EF1-Restoration problem is in **PSPACE**. Recall that **PSPACE** is the set of all decision problems that can be solved by a deterministic polynomial-space Turing machine. We can solve the EF1-Restoration problem non-deterministically by simply guessing a path from the given near-EF1 allocation to some EF1 allocation. Since the total number of allocations is at most n^m , if there exists such a path, then there exists one with length at most n^m. This shows that the problem is in **NPSPACE**, the non-deterministic analogue of **PSPACE**.[2] It is known that **NPSPACE** = **PSPACE** [32], which implies that the EF1-Restoration problem is in **PSPACE**. We refer the reader to a standard complexity theory text like [4] for formal definitions of **PSPACE**, **NPSPACE**, etc.

PSPACE-*hardness:* We reduce the **PSPACE**-complete problem of Perfect Matching Reconfiguration problem to EF1-Restoration.

Definition 6.1 (The Perfect Matching Reconfiguration problem)**.** For an undirected bipartite graph $G = (A \sqcup B, E)$ with $|A| = |B| = n$, the problem is to decide reachability between two given perfect matchings M_0 and M^*. It involves deciding if M_0 and M^* can be reached from each other via a sequence of perfect matchings $M_0, M_1, M_2, \dots, M_t = M^*$, such that for each $k \in [t]$, there exist edges $e_1^k, e_2^k, e_3^k, e_4^k$ of G such that $M_{k-1} \setminus M_k = \{e_1^k, e_3^k\}$, $M_k \setminus M_{k-1} = \{e_2^k, e_4^k\}$, and $e_1^k, e_2^k, e_3^k, e_4^k$ form a cycle.

The operation of going from M_{k-1} to M_k is called a *flip*, and we say that M_{k-1} and M_k are *adjacent* to each other.

Construction of an instance of the EF1-Restoration *problem:* For a given instance of Perfect Matching Reconfiguration as described above, let us denote $A = \{a_1, a_2, \dots, a_n\}$ and $B = \{b_1, b_2, \dots, b_n\}$. We write $N(v)$ to denote the set of neighbors of vertex $v \in A \cup B$ in G. By possible renaming of vertices, let $M_0 = \{(a_i, b_i)\}_{i \in [n]}$ and the final matching $M^* = \{(a_i, b_{\pi(i)})\}_{i \in [n]}$, where π is a permutation on $[n]$.

Now, we construct an instance of the EF1-Restoration problem, with agents having monotone binary valuations.

- Create a set $\mathcal{N} = \{0, \dots, n+2\}$ of $n+3$ agents and a set $\mathcal{G} = \{a_i, \bar{a}_i, b_i \mid i \in [n]\} \cup \{r_1, r_2, r_3, r_4\}$ of $3n+4$ items. That is, for each $i \in [n]$, we have three items $a_i, \bar{a}_i$, and b_i, with four additional items r_1, r_2, r_3, and r_4.
- Create the initial near-EF1 allocation $X = \{X_0, \dots, X_{n+2}\}$, that is given as an input, from the matching M_0 of G as $X_0 := \emptyset$, $X_i := \{a_i, \bar{a}_i, b_i\}$ for each $i \in [n]$, $X_{n+1} := \{r_1, r_2\}$, and $X_{n+2} := \{r_3, r_4\}$.
- We define the valuations as follows:

$$v_0(S) := \begin{cases} 1 & \text{if } S = \left\{a_i, \bar{a}_i, b_j\right\} \ \forall i, j \in [n], \\ 1 & \text{if } S \subsetneq \left\{a_i, \bar{a}_i, b_j\right\}, \text{ with } |S| = 2, \\ & \quad \forall i \in [n],\ \forall j \in [n] \setminus \pi(i), \\ 0 & \text{if } S = \left\{a_i/\bar{a}_i, b_{\pi(i)}\right\},\ \forall i \in [n], \\ 0 & \text{otherwise.} \end{cases}$$

For each agent $i \in [n]$, we have $X_i = \{a_i, \bar{a}_i, b_i\}$ in X. We define the valuation for agent $i \in [n]$ as

$$v_i(S) := \begin{cases} 1 & \text{if } S = \left\{a_i, \bar{a}_i, b_j\right\} \ \forall i \in [n], \forall b_j \in N(a_i), \\ 0 & \text{if } S \subsetneq \left\{a_i, \bar{a}_i, b_j\right\} \ \forall i \in [n], \forall b_j \in N(a_i), \\ 1 & \text{if } S = \{r_3\} \ \text{ or } S = \{r_4\}, \\ 0 & \text{if } S = \left\{a_i/\bar{a}_i, b_j, c\right\}, \forall c \in \{b_k, a_k, \bar{a}_k, r_1, r_2\}, \\ & \quad \forall i, k \in [n], \forall b_j \in N(a_i), \\ 0 & \text{otherwise.} \end{cases}$$

The valuation of agent $n+1$, where $X_{n+1} = \{r_1, r_2\}$ in X, is defined below.

[2] **NPSPACE** is the set of all decision problems that can be solved by a non-deterministic polynomial-space Turing machine.

$$v_{n+1}(S) := \begin{cases} 1 & \text{if } S = \{r_1, r_2\}, \\ 0 & \text{if } S = \{r_1\} \text{ or } S = \{r_2\}, \\ 0 & \text{if } S = \{r_1/r_2, c\}, \forall c \in \{r_3, r_4, b_i\}, \ \forall i \in [n], \\ 1 & \text{if } S \subsetneq \{a_i, \bar{a}_i, b_j\}, \text{ with } |S| = 2, \\ & \quad \forall i \in [n], \ \forall j \in [n] \setminus \pi(i), \\ 0 & \text{if } S = \{a_i/\bar{a}_i, b_{\pi(i)}\}, \\ 0 & \text{otherwise.} \end{cases}$$

We define the valuation of agent $n+2$ as follows, who holds the bundle $X_{n+2} = \{r_3, r_4\}$ in X.

$$v_{n+2}(S) := \begin{cases} 1 & \text{if } S = \{r_3, r_4\}, \\ 0 & \text{if } S = \{r_3\} \text{ or } S = \{r_4\}, \\ 1 & \text{if } S = \{r_1\} \text{ or } S = \{r_2\}, \\ 0 & \text{if } S = \{r_3/r_4, c\}, \forall c \in \{a_i, \bar{a}_i, b_i\}, \ \forall i \in [n], \\ 0 & \text{otherwise.} \end{cases}$$

Note that the valuations are listed above explicitly for a polynomial number of sets. We extend the valuations to other non-listed subsets monotonically. Clearly, this instance of EF1-Restoration can be constructed in polynomial time.

Properties of X: Note that in the initial allocation X, agent 0 has EF1-envy towards all the agents $i \in [n]$ for whom $b_i \neq b_{\pi(i)}$. Agent 0 is the unhappy agent in all of the near-EF1 allocations in our instance. Observe that the initial allocation X is near-EF1, as no agent apart from agent 0 has EF1-envy towards any other agent. The goal is to perform a sequence of operations, i.e., valid transfers or exchanges, and finally reach some EF1 allocation.

Note that there is only one EF1 allocation Z in our instance, where $Z_i = \{a_i, \bar{a}_i, b_{\pi(i)}\}$ for each $i \in [n]$, $Z_0 = \emptyset$, $Z_{n+1} = X_{n+1}$, and $Z_{n+2} = X_{n+2}$. These bundles correspond to M^* in the Perfect Matching Reconfiguration problem.

The valuations forbid the following types of operations:

- *No agent $i \in \mathcal{N}$ can transfer an item from her bundle*, because she would then EF1-envy either agent $n+2$ (if $i \neq n+2$), or agent $n+1$ (if $i = n+2$).
- *No agent $i \in [n]$ can exchange an item with agent $n+1$*. If agent i gives away a_i or $\bar{a}_i$ in exchange of r_1 or r_2, then she EF1-envies agent $n+2$. Also, if $b_j \in X_i$ is exchanged with agent $n+1$ for r_1 or r_2, then the value of agent $n+1$'s bundle drops to 0 and she EF1-envies each agent $j \in [n]$ who does not possess $b_{\pi(j)}$.
- *No agent $i \in [n]$ can exchange the a_i or $\bar{a}_i$ in their bundle, with any agent j* because agent i would then EF1-envy agent $n+2$.
- *No agent $i \in [n]$ can exchange any pair of items with agent $n+2$*, since agent $n+2$ would then EF1-envy agent $n+1$.
- *No agent $i \in [n]$ can exchange the b_j in their current bundle with some other $b_{j'}$, if $b_{j'} \notin N(a_i)$*, for agent i would then EF1-envy agent $n+2$.
- *Agent $n+1$ can not exchange any pair of items with agent $n+2$*, for agent $n+1$ would then EF1-envy each agent $i \in [n]$ who does not possess $b_{\pi(i)}$.
- *Agent 0 cannot receive any item.* This is because agent 0 does not value the items r_1, r_2, r_3, r_4, and agent $i \in [n]$ cannot transfer an item as stated above.

Therefore, the only valid operation is an exchange of the *b-type* items between a pair of agents i, j. More precisely, let the near-EF1 allocation at some point be Y, such that $Y_i = \{a_i, \bar{a}_i, b_k\}$ and $Y_j = \{a_j, \bar{a}_j, b_\ell\}$. The only valid operation is an exchange of b_k and b_ℓ between agents i, j.

Suppose, at some point, we have an allocation Y where $Y_i = \{a_i, \bar{a}_i, b_k\}$ and $Y_j = \{a_j, \bar{a}_j, b_\ell\}$ for some $i, j \in [n]$. This represents (a part of) a perfect matching M' in the graph G, where $(a_i, b_k), (a_j, b_\ell) \in M'$. An exchange of the items b_k and b_ℓ between agents i and j corresponds to a *flip* of the perfect matching M' so that the edges $(a_i, b_k), (a_j, b_\ell)$ are replaced by (a_i, b_ℓ) and (a_j, b_k) in the new perfect matching M''. Therefore, every valid operation in our instance corresponds to a flip in the perfect matching in the given instance. Moreover, for any intermediate near-EF1 allocation Y, we may simply obtain the corresponding perfect matching as follows - vertex a_i is matched to vertex b_j if $b_j \in Y_i$ for $i \in [n]$.

Over the course of these valid operations, the unhappy agent 0 will receive no item, and continue to hold an empty bundle. Therefore, the only way to make agent 0 lose their EF1-envy is to reach an allocation X^* in which each agent $i \in [n]$ holds the bundle $X_i^* = \{a_i, \bar{a}_i, b_{\pi(i)}\}$. Observe that such an allocation X^* corresponds to the perfect matching where there exists an edge between vertices a_i and $b_{\pi(i)}$ for each $i \in [n]$. This is precisely the target perfect matching M^*.

Thus, there is a sequence of adjacent perfect matchings from M_0 to M^* in G if and only if there is a sequence of valid operations that transforms X to the unique EF1 allocation Z. This completes the reduction. □

7 FUTURE DIRECTIONS

The meta-question raised by this work is the following: what is/are the "least complicated valid operation(s)" that one has to permit to reach a *fair* allocation by never breaking the *near-fair* guarantees? This leads to several natural open directions that one might explore, and we mention a few.

(1) Tractability of EF1-Restoration for *additive* binary valuations, and EF1-Restoration in the case of orientations under additive/monotone binary valuations - can we remove our additional requirement of the valuations being graphical?
(2) Analogs of EF1-Restoration for other notions of fairness, and valuation classes - EF1 for mixed manna, EFX for identical/ordered valuations, PROP1, etc.
(3) A variant of the problem in which the valuations change as a part of the changing environment.

REFERENCES

[1] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online fair division: analysing a food bank problem. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*. AAAI Press, Buenos Aires, Argentina, 2540–2546.

[2] Martin Aleksandrov and Toby Walsh. 2017. Pure Nash Equilibria in Online Fair Division. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, Melbourne, Australia, 42–48.

[3] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. 2023. Fair division of

indivisible goods: Recent progress and open questions. *Artificial Intelligence* 322 (2023), 103965.
[4] Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity - A Modern Approach.* Cambridge University Press, Cambridge, UK.
[5] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. 2019. Fair Allocation of Indivisible Goods and Chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI).* ijcai.org, Macao, China, 53–59.
[6] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Finding Fair and Efficient Allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC).* ACM, Ithaca, NY, USA, 557–574.
[7] Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. 2021. On Approximate Envy-Freeness for Indivisible Chores and Mixed Resources. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM).* Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Seattle, WA, USA, 1:1–1:23.
[8] Arpita Biswas, Justin Payan, Rik Sengupta, and Vignesh Viswanathan. 2023. *The Theory of Fair Allocation Under Structured Set Constraints.* Springer Nature Singapore, Singapore, 115–129.
[9] Marthe Bonamy, Nicolas Bousquet, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Arnaud Mary, Moritz Mühlenthaler, and Kunihiro Wasa. 2019. The Perfect Matching Reconfiguration Problem. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS).* Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Aachen, Germany, 80:1–80:14.
[10] Steven J. Brams and Alan D. Taylor. 1996. *Fair division - from cake-cutting to dispute resolution.* Cambridge University Press, Cambridge, UK.
[11] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. 2016. *Handbook of Computational Social Choice.* Cambridge University Press, Cambridge, UK.
[12] Eric Budish. 2011. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.
[13] Eric Budish and Estelle Cantillon. 2007. Strategic Behavior in Multi-unit Assignment Problems: Theory and Evidence from Course Allocations. In *Computational Social Systems and the Internet, 1.7. - 6.7.2007 (Dagstuhl Seminar Proceedings, Vol. 07271).* Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 1–1.
[14] Mithun Chakraborty, Ayumi Igarashi, Warut Suksompong, and Yair Zick. 2021. Weighted Envy-freeness in Indivisible Item Allocation. *ACM Trans. Economics and Comput.* 9, 3 (2021), 18:1–18:39.
[15] Mithun Chakraborty, Ulrike Schmidt-Kraepelin, and Warut Suksompong. 2021. Picking sequences and monotonicity in weighted fair division. *Artif. Intell.* 301 (2021), 103578.
[16] Harish Chandramouleeswaran, Prajakta Nimbhorkar, and Nidhi Rathi. 2024. Fair Division in a Variable Setting. arXiv:2410.14421 [cs.GT] https://arxiv.org/abs/2410.14421
[17] George Christodoulou, Amos Fiat, Elias Koutsoupias, and Alkmini Sgouritsa. 2023. Fair allocation in graphs. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC).* ACM, London, UK, 473–488.
[18] Argyrios Deligkas, Eduard Eiben, Tiger-Lily Goldsmith, and Viktoriia Korchemna. 2024. EF1 and EFX Orientations. arXiv:2409.13616 [cs.GT] https://arxiv.org/abs/2409.13616
[19] Raúl H. Etkin, Abhay Parekh, and David Tse. 2007. Spectrum sharing for unlicensed bands. *IEEE J. Sel. Areas Commun.* 25, 3 (2007), 517–528.
[20] Duncan Karl Foley. 1966. *Resource allocation and the public sector.* Vol. 7:45-98. Yale Economic Essays, New Haven, CT, USA.
[21] Hao Guo, Weidong Li, and Bin Deng. 2023. A survey on fair allocation of chores. *Mathematics* 11, 16 (2023), 3616.
[22] Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. 2020. Fair Division with Binary Valuations: One Rule to Rule Them All. In *Web and Internet Economics - 16th International Conference, WINE (Lecture Notes in Computer Science, Vol. 12495).* Springer, Beijing, China, 370–383.
[23] Jiafan He, Ariel D. Procaccia, Alexandros Psomas, and David Zeng. 2019. Achieving a Fairer Future by Changing the Past. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, Sarit Kraus (Ed.). ijcai.org, Macao, China, 343–349.
[24] Ayumi Igarashi, Naoyuki Kamiyama, Warut Suksompong, and Sheung Man Yuen. 2024. Reachability of Fair Allocations via Sequential Exchanges. *Algorithmica* 86, 12 (2024), 3653–3683.
[25] Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. 2023. On reachable assignments under dichotomous preferences. *Theor. Comput. Sci.* 979 (2023), 114196.
[26] Ian A. Kash, Ariel D. Procaccia, and Nisarg Shah. 2014. No Agent Left Behind: Dynamic Fair Division of Multiple Resources. *J. Artif. Intell. Res.* 51 (2014), 579–603.
[27] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC).* ACM, New York, NY, USA, 125–131.
[28] Nicholas Mattei, Abdallah Saffidine, and Toby Walsh. 2017. Mechanisms for Online Organ Matching. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI).* ijcai.org, Melbourne, Australia, 345–351.
[29] Hervé Moulin. 2003. *Fair division and collective welfare.* MIT Press, Cambridge, MA, USA.
[30] John Winsor Pratt and Richard Jay Zeckhauser. 1990. The fair and efficient division of the Winsor family silver. *Management Science* 36, 11 (1990), 1293–1301.
[31] Jack M. Robertson and William A. Webb. 1998. *Cake-cutting algorithms - be fair if you can.* A K Peters, Wellesley, MA, USA.
[32] Walter J. Savitch. 1970. Relationships Between Nondeterministic and Deterministic Tape Complexities. *J. Comput. Syst. Sci.* 4, 2 (1970), 177–192.
[33] Erel Segal-Halevi and Balázs R. Sziklai. 2018. Resource-monotonicity and population-monotonicity in connected cake-cutting. *Math. Soc. Sci.* 95 (2018), 19–30.
[34] Erel Segal-Halevi and Balázs R. Sziklai. 2019. Monotonicity and competitive equilibrium in cake-cutting. *Economic Theory* 68, 2 (2019), 363–401.
[35] Hugo Steinhaus. 1948. The problem of fair division. *Econometrica* 16 (1948), 101–104.
[36] Walter Stromquist. 1980. How to Cut a Cake Fairly. *The American Mathematical Monthly* 87, 8 (1980), 640–644.
[37] Francis Edward Su. 1999. Rental Harmony: Sperner's Lemma in Fair Division. *The American Mathematical Monthly* 106, 10 (1999), 930–942.
[38] Thomas Vossen. 2002. *Fair Allocation Concepts in Air Traffic Management.* Ph.D. Dissertation. University of Maryland.
[39] Toby Walsh. 2011. Online Cake Cutting. In *Algorithmic Decision Theory - 2nd International Conference (ADT) (Lecture Notes in Computer Science, Vol. 6992).* Springer, Piscataway, NJ, USA, 292–305.
[40] Jinghan A Zeng and Ruta Mehta. 2024. On the structure of EFX orientations on graphs. arXiv:2404.13527 [cs.GT] https://arxiv.org/abs/2404.13527
[41] Yu Zhou, Tianze Wei, Minming Li, and Bo Li. 2024. A Complete Landscape of EFX Allocations on Graphs: Goods, Chores and Mixed Manna. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI).* ijcai.org, Jeju, South Korea, 3049–3056.