Temporal Fair Division

Benjamin Cookson , **Soroush Ebadian** , **Nisarg Shah**
University of Toronto
{bcookson,soroush,nisarg}@cs.toronto.edu

Abstract

We study temporal fair division, whereby a set of agents are allocated a (possibly different) set of goods on each day for a period of days. We study this setting, as well as a number of its special cases formed by the restrictions to two agents, same goods on each day, identical preferences, or combinations thereof, and chart out the landscape of achieving two types of fairness guarantees simultaneously: fairness on each day (per day) and fairness over time (up to each day, or the weaker version, overall).

In the most general setting, we prove that there always exists an allocation that is stochastically-dominant envy-free up to one good (SD-EF1) per day and proportional up to one good (PROP1) overall, and when all the agents have identical preferences, we show that SD-EF1 per day and SD-EF1 overall can be guaranteed. For the case of two agents, we prove that SD-EF1 per day and EF1 up to each day can be guaranteed using an envy balancing technique. We provide counterexamples for other combinations that establish our results as among the best guarantees possible, but also leave open some tantalizing questions.

1 Introduction

How to divide a set of goods amongst a set of agents *fairly* has been an enigma for centuries. There has been remarkable progress on this question in the last decade [Amanatidis *et al.*, 2022]. In the most prominent model, there is a set of n agents N, each having an (additive) valuation over a set of goods M. The goal is to find an allocation $A = (A_1, \ldots, A_n)$ which partitions M into pairwise-disjoint bundles, one allocated to each agent $i \in N$.

This one-shot model fails to capture numerous real-world fair division scenarios in which goods are divided over time, e.g., food bank deliveries [Lee *et al.*, 2019], resource allocation in data centers [Ghodsi *et al.*, 2013], allocation of advertising slots [Mehta *et al.*, 2007], nurse shift scheduling [Miller *et al.*, 1976], and organ transplants [Bertsimas *et al.*, 2013]. Compared to the one-shot setting, fair division of goods over time has received relatively little attention.

Inspired by this, there has been a flurry of recent works that consider *online fair division*, where agents or goods arrive over time and the principal needs to make allocations in an online fashion in the absence of any information regarding future arrivals [Kash *et al.*, 2014; Benadè *et al.*, 2024]. The limits of feasible fairness guarantees have been explored under various adversary models [Zeng and Psomas, 2020; Benadè *et al.*, 2024].

However, in practice it is rarely the case that we have absolutely no information about the future. Significantly better guarantees have been established when even partial information about the future is available, either in the form of distributional knowledge [Bogomolnaia *et al.*, 2022] or machine-generated predictions [Gkatzelis *et al.*, 2021; Banerjee *et al.*, 2022, 2023]. But this work has left a very basic question wide open: *How fair can we be if we had full information about the future?*

To address this, we introduce the model of *temporal fair division*, where a set of agents N are allocated a set of goods M_t on day t, over a period of days $t \in \{1, \ldots, k\}$, and the agents' valuations over the whole set of goods $M = \cup_{t=1}^{k} M_t$ are available upfront. At first glance, it may seem that this is just a traditional fair division problem where the set of goods M needs to be divided amongst the set of agents N. The twist, however, is that in temporal fair division, agents anticipate fairness to prevail not solely at the end of the entire time horizon, but also at or within various interim time intervals. For example, the principal may be confident, based on their knowledge of the future, the allocation will eventually turn out to be fair, but that may not be assurance enough to the agents.

This leads us to seek *temporal fairness notions* in our temporal fair division setting. Specifically, we take prominent fairness notions from one-shot fair division, and seek them on three temporal scales:

(1) *Per day:* The allocation of the set of goods M_t on each day t should be fair.

(2a) *Overall:* The allocation of the whole set of goods M in the end should be fair.

(2b) *Up to each day:* The allocation of the set of goods $\cup_{r=1}^{t} M_r$ up to each day t should be fair.

Clearly, up to each day fairness (2b) is stronger than overall fairness (2a). Solely achieving per day fairness (1) or overall

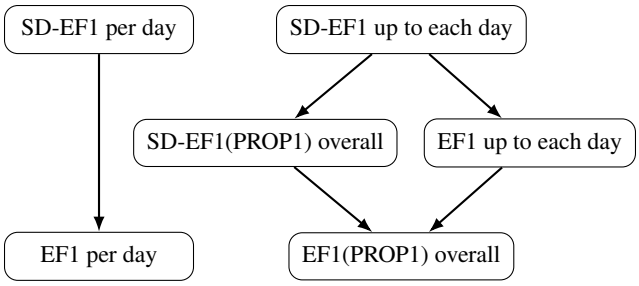

Figure 1: Hierarchy of temporal fairness notions.

fairness (2a) can be reduced to one-shot fair division. Hence, we seek per day and overall fairness simultaneously (1+2a), or per day and up to each day fairness simultaneously (1+2b), or solely up to each day fairness (2b). Our main research question is to...

> *...explore the limits of temporal fairness that can be guaranteed in temporal fair division.*

1.1 Our Results & Techniques

We chart out the landscape of the aforementioned temporal fair division model in a general setting, with n agents having additive, heterogeneous preferences. Further, we identify three relevant restrictions where we can circumvent some of the impossibilities of the general setting, and achieve very strong results. Those are: (1) when there are only two agents; (2) when all agents have the same ordering over the goods; and (3) when an identical set of goods arrives each day.

In these settings, we seek the fairness guarantees of EF1, SD-EF1, PROP1, and SD-PROP1 at the temporal scales of per day, overall, and up to each day. The various temporal fairness definitions are depicted in Figure 1, with arrows indicating logical implications. We discover several surprising results, and develop novel algorithmic tools along the way, which may be of independent interest.

In Section 3, we present an algorithm for finding temporally fair allocations in our most general setting. Specifically, we show how to obtain an allocation that is PROP1 overall, and SD-EF1 per day in polynomial time (Theorem 1).

In Sections 4 to 6, we look at the restricted settings of two agents, identical orderings, and identical days, respectively. In these settings, we provide algorithms that give very strong fairness guarantees that are impossible in the general case (Theorems 3, 7 and 9), while also showing which fairness desiderata are still too strong even after applying these restrictions (Theorems 4, 5 and 8). In these restricted settings, we provide a near complete picture of what is possible, while leaving open some interesting questions, particularly surrounding the powerful notion of "up to each day" fairness.

Most of our results, along with several open questions, are summarized in Table 1.

1.2 Related Work

Our temporal fair division model is related to (but separate from) several fair division models studied in the literature. We include the most relevant ones here, and include a more in-depth look at the literature in the Appendix A

Repeated fair division. The repeated fair division model of Igarashi *et al.* [2024], which is the case of identical days in our more general model, is the most closely related to our work. Some of their results are for two agents (still with identical days). As will be seen in Section 4, this is where the strongest guarantees from Figure 1 of SD-EF1 per day and SD-EF1 up to each day can be achieved simultaneously. This result is the only overlap between our works, and we present it again because we obtain a shorter proof with a much simpler algorithm. The rest of their results with two or more than two agents seek exact fairness guarantees, such as (exact) envy-freeness overall, in limited cases such as when the number of days is a multiple of the number of agents.

Online fair division. In the online fair division model, goods arrive one by one and must be irrevocably allocated to an agent upon arrival with no knowledge of agent preferences over the goods to arrive later. Typically, one seeks to maintain a certain level of fairness. Clearly, any online fair division algorithm can be simulated in our temporal fair division model to achieve the same guarantee up to each day. One online fair division paper of particular note to this research is He *et al.* [2019]. This paper introduced the "Informed Model" of online fair division, where irrevocable allocation decisions must be made as goods arrive one at a time in adversarial order, but the allocation algorithm is given all goods and the order they will arrive in advance. The main goal in the informed model is to achieve an allocation that remains EF1 after each good is allocated. Clearly, this is equivalent to achieving EF1 up to each day in Temporal Fair Division when each day only contains a single good. He *et al.* concludes that it is impossible to allocate the goods in such a way that EF1 is always maintained. By corollary, EF1 up to each day is also infeasible.

Constrained fair division. Seeking fair allocations over a set of goods that remain fair when only looking at the goods from a single "day" can also be modeled as a constraint on the space of feasible allocations, and the question becomes whether there is a *constrained allocation* that still achieves the desired fairness guarantee. This model of constrained fair division has also been studied in the literature. Biswas and Barman [2018] study a model with cardinality constraints. Cardinality constraints (partition matroid constraints) have been generalized to matroid constraints, and the existence of an EF1 allocation subject to matroid feasibility constraints is a major open question [Biswas and Barman, 2018; Dror *et al.*, 2023]. Finally, the bihierarchy framework of Budish *et al.* [2013] can also be viewed as a method for finding a constrained allocation, which we use in some of our results. Although, our most interesting results deal with sets of constraints that go beyond bihierarchies.

Temporal fairness in social choice. While we look at temporal fairness applied to the allocation of indivisible resources, the idea of temporal fairness has been explored in other areas of social choice theory. See the work of Elkind *et al.* [2024b] for a detailed look at temporal multi-winner voting, and for a synopsis of other papers that look at fairness over time. Also, Alamdari *et al.* [2024] present a model of temporal fairness for a very general decision making setting.

	Up to each day		Overall			
	SD-EF1	EF1	SD-EF1	SD-PROP1	EF1	PROP1
General Setting						
SD-EF1 Per Day	X	X	X	?	?	✓ (Thm 1)
EF1 Per Day	X	X	X	?	?	✓
$\emptyset$	X	X [He *et al.*, 2019]	✓ [Aziz, 2020]	✓	✓	✓
Two Agents						
SD-EF1 Per Day	X	✓(Thm 3)	X	?	✓	✓
EF1 Per Day	X	✓	X (Thm 5)	?	✓	✓
$\emptyset$	X (Thm 4)	✓	✓	✓	✓	✓
Identical Orderings						
SD-EF1 Per Day	X	?	✓(Thm 7)	✓	✓	✓
EF1 Per Day	X	?	✓	✓	✓	✓
$\emptyset$	X (Thm 8)	?	✓	✓	✓	✓
Identical Days						
SD-EF1 Per Day	X	?	?	✓ (Thm 9)	?	✓
EF1 Per Day	X	?	?	✓	?	✓
$\emptyset$	X (Thm 8)	?	✓	✓	✓	✓

Table 1: Possibilities, impossibilities, and open questions in temporal fair division. ✓indicates a possibility result. X indicates an impossibility. ? indicates an open question. Green highlights indicate the main results of this paper; non-highlighted cells are either open, already known, or implied by other results.

Finally, contemporarily to and independently of our work, Elkind *et al.* [2024a] also study the same temporal fair division model as ours, but their results have no overlap with ours. In particular, they focus primarily on EF1 up to each day, achieving it for several special cases of restricted instances.

2 Preliminaries

2.1 Model

For any $r \in \mathbb{N}$, define $[r] \triangleq \{1, 2, \ldots, r\}$. A multiset is a set that allows repetitions.

Agents, goods, and valuations. Let $N = [n]$ be a set of agents who are allocated a set of goods on each day over k consecutive days. For $t \in [k]$, denote by M_t the set of goods to be allocated on day t, $\overline{M}_t = \cup_{r \in [t]} M_r$ the set of goods up to day t, and $M = \overline{M}_k = \cup_{t \in [k]} M_t$ the set of all goods. We can view $(M_1, \ldots, M_k)$ as a partition of M. Each agent $i \in N$ has an additive valuation function $v_i : 2^M \to \mathbb{R}_{\geqslant 0}$, where $v_i(\{g\})$ (henceforth, with a slight abuse of notation, written as $v_i(g)$) is her utility for receiving good $g \in M$ and $v_i(S) = \sum_{g \in S} v_i(g)$ for all $S \subseteq M$. Collectively, $(N, (M_1, \ldots, M_k), \{v_i\}_{i \in N})$ form an instance of *temporal fair division*. An instance with $k = 1$ is a (regular) fair division instance, so a temporal fair division instance can be viewed as a sequence of fair division instances in which the same agents participate. We will always assume that an algorithm to solve a temporal fair division problem is given the entire instance as input, i.e. it knows the entire set of goods and what day those goods will arrive upfront.

Preferences. Define $\succcurlyeq_i$ (resp., $\succ_i$) as the weak (resp., strict) ordering over the goods in M induced by v_i, where, for all $g, g' \in M$, $g \succcurlyeq_i g'$ if and only if $v_i(g) \geqslant v_i(g')$ and $g \succ_i g'$ if and only if $v_i(g) > v_i(g')$. For all $S \subseteq M$ and all $r \in \mathbb{N}$, define $T_i(S, r)$ to be the r most preferred goods among the goods in S according to the ordering $\succcurlyeq_i$; all ties are broken consistently across i, S, and r.[1]

Allocations. An allocation $A = (A_1, \ldots, A_n)$ is a partition of M into n pairwise-disjoint bundles, where A_i is the bundle allocated to agent i. For $S \subseteq M$, let $A_S = (A_{S,1}, \ldots, A_{S,n})$ be the allocation of the goods in S that is induced by A (i.e., for each good $g \in M$ and agent $i \in N$, $g \in A_{S,i}$ if and only if $g \in S$ and $g \in A_i$). For $t \in [k]$, we refer to A_{M_t} as the allocation on day t and $A_{\overline{M}_t}$ as the allocation up to day t.

Restrictions. We study three restrictions of this general setup (and their combinations).

1. *Two agents:* $|N| = 2$.
2. *Identical valuations/orderings:* Under identical valuations, $v_i = v$ for all agents $i \in N$. Under identical orderings, $\succcurlyeq_i = \succcurlyeq$ for all agents $i \in N$. Here, we simply write v, $\succcurlyeq$, and $T(S, r)$, skipping the agent in the subscript. For this case, our results deal with desiderata which depend only on the orderings; thus, no distinction between valuations and orderings is necessary.[2]

[1] That is, we use an arbitrary global ordering over M as the tiebreaker to convert the weak ordering $\succcurlyeq_i$ of every agent $i \in N$ into her strict ordering over M, and compute all $T_i(S, r)$-s according to these strict orderings. Our negative results do not depend on this tie-breaking and positive results hold regardless of it.

[2] In other words, our positive results hold even under identical orderings (weaker restriction), while our negative results hold even under identical valuations (stronger restriction).

3. *Identical days:* Informally, copies of the same goods are allocated on each day. Formally, for all days $t, t' \in [k]$, there is a bijection $f : M_t \to M_{t'}$ such that $v_i(g) = v_i(f(g))$ for all agents $i \in N$ and goods $g \in M_t$.

2.2 Fairness Desiderata

We first introduce the main desiderata we will be studying. Later, we will introduce their temporal extensions. Other notions referred to in specific sections will be introduced therein.

Definition 1 (Envy-Freeness Up to One Good (EF1))**.** *An allocation A of a set of goods S is envy-free up to one good (EF1) if for all $i, j \in N$ with $A_j \neq \emptyset$, there exists a $g \in A_j$ such that $v_i(A_i) \geqslant v_i(A_j \setminus \{g\})$, i.e., no agent envies another agent if some good from the latter agent's bundle is removed.*

In addition to EF1, we will also introduce a weaker notion of measuring fairness that does not require directly comparing one agent's bundle to another's.

Definition 2 (Proportionality Up to One Good (PROP1))**.** *An allocation A of a set of goods S is proportional up to one good (PROP1) if for all $i \in N$ with $A_i \neq S$, there exists a good $g \in S \setminus A_i$, such that $v_i(A_i \cup \{g\}) \geqslant \frac{1}{n} v_i(S)$.*

It is well known that EF1 is a stronger notion than PROP1 [Conitzer *et al.*, 2017].

Given a set of goods S and a good $g \in S$, define an agent i's top-set with respect to S as $H_i(S, g) = \{g' \in S : g' \succcurlyeq_i g\}$. When given only a weak ordering $\succcurlyeq_i$ over a set of goods S, we can compare two bundles $X, Y \subseteq S$ using the *stochastic dominance* (SD) relation: $X \succcurlyeq_i^{\text{SD}} Y$ if for all $g \in S$, $|X \cap H_i(S, g)| \geqslant |Y \cap H_i(S, g)|$. That is, X has at least as many goods weakly preferred to any good as Y has. It is known that $X \succcurlyeq_i^{\text{SD}} Y$ if and only if $v_i(X) \geqslant v_i(Y)$ for all (additive) valuations v_i over S that would induce $\succcurlyeq_i$. Hence, using the SD comparison in the EF1 definition yields its stronger counterpart, which has also been studied extensively [Aziz *et al.*, 2015; Freeman *et al.*, 2021; Aziz *et al.*, 2023].

Definition 3 (SD-EF1)**.** *An allocation A of a set of goods S is stochastically-dominant envy-free up to one good (SD-EF1) if for all $i, j \in N$ with $A_j \neq \emptyset$, there exists a $g \in A_j$ such that $A_i \succcurlyeq_i^{SD} A_j \setminus \{g\}$.*

We also introduce a stochastic dominance extension of PROP1.

Definition 4 (SD-PROP1)**.** *An allocation A of a set of goods S is stochastically-dominant proportional up to one good (SD-PROP1) if, for all $i \in N$ with $A_i \neq S$, there exists a $g \in S \setminus A_i$ such that $|(A_i \cup \{g\}) \cap H_i(S, g')| \geqslant \lceil {}^{|H_i(S,g')|}/_{n} \rceil$ for all $g' \in S$.*

Informally, A_i, after adding at most one good to it, must contain at least $\lceil {}^{k}/_{n} \rceil$ goods among the k most preferred goods of agent i in S, for each $k \in [|S|]$.

Just as EF1 implies PROP1, we have that SD-EF1 implies SD-PROP1, and similarly to SD-EF1, if an allocation A is SD-PROP1 for certain orderings $\{\succcurlyeq_i\}_{i \in N}$, then A will be PROP1 for any additive valuation functions that induce $\{\succcurlyeq_i\}_{i \in N}$. Both these facts are proven in the Appendix B.

2.3 Temporal Fairness

In a temporal fair division instance given by a set of goods M partitioned as $(M_1, \ldots, M_k)$ across k days, we can ask for fairness to hold at different levels of granularity, yielding various temporal extensions of the fairness desiderata introduced above. These extensions also apply to any other type of desiderata (e.g., efficiency).

Definition 5 (Per Day Fairness)**.** *For desideratum X, allocation A satisfies X per day if A_{M_t} satisfies X for all $t \in [k]$.*

Definition 6 (Overall Fairness)**.** *For desideratum X, allocation A satisfies X* overall *if $A_M = A$ satisfies X.*

Definition 7 (Up To Each Day Fairness)**.** *For desideratum X, allocation A satisfies X up to each day if $A_{\overline{M}_t}$ satisfies X for all $t \in [k]$.*

Note that 'up to each day' is a strengthening of 'overall', while 'per day' is incomparable to those two. Plugging in our fairness desiderata into these three temporal extensions gives us the hierarchy of fairness guarantees depicted in Figure 1. Because SD-EF1 is achievable for (regular) fair division (e.g., via a simple round-robin procedure [Caragiannis *et al.*, 2019]), SD-EF1 per day and SD-EF1 overall are both individually achievable, implying the same for EF1, PROP1, and SD-PROP1.

3 General Preferences

In this section, we present temporal fair division results in the most general setting: an arbitrary set of goods arrive each day, and each agent has arbitrary additive preferences over them. Let us present our main result for this general setting.

Theorem 1. *For any temporal fair division instance, an allocation that is SD-EF1 per day and PROP1 overall exists and can be computed in polynomial time.*

We find such an allocation using Algorithm 1. The derivation of Theorem 1 can be divided into three conceptual steps.

Identical ordering transformation. Algorithm 1 begins by creating an auxiliary temporal fair division instance as follows. For each day t, it creates a new instance for that day with a set of goods M'_t and valuations v' such that agents have identical orderings but with the same set of utility values as they had previously. More formally, let $M'_t = \{g'_{t,1}, \ldots, g'_{t,|M_t|}\}$. Then, for each agent $i \in N$, $v'_i(g'_{t,1}) \geqslant v'_i(g'_{t,2}) \ldots \geqslant v'_i(g'_{t,|M'_t|})$ (common ordering) and there exists a bijection $o_{i,t}$ between M_t and M'_t such that $v_i(g) = v'_i(o_{i,t}(g))$ for all $g \in M_t$ (same utility values). This technique has been used previously for designing algorithms to achieve (approximate) maximin share fairness (MMS) [Bouveret and Lemaître, 2014], but we use it with a novel and nontrivial analysis to ensure PROP1.

Connection to cardinality constraints. Algorithm 1 invokes a key subroutine due to Biswas and Barman [2018] that returns an EF1 allocation subject to cardinality constraints summarized below.

Theorem 2 (Theorem 1 of [Biswas and Barman, 2018])**.** *Given p disjoint sets of goods $C_1, \ldots, C_p$ and n agents with*

Algorithm 1 SD-EF1 per day + PROP1 Overall

// Identical Ordering Transformation
$v'_i = \emptyset$ for all $i \in N$
for $t \in [k]$ **do**
  $M'_t \leftarrow \{g'_{t,1}, \ldots, g'_{t,|M_t|}\}$
  for $i \in N$ **do**
    $o_{i,t}$ be the goods M_t in non-increasing order of v_i
    $v'_i(g'_{t,j}) \leftarrow v_i(o_{i,t}(j))$ for all $j \in [|M_t|]$
  end for
end for

// Invoking EF1 with Cardinality Constraints algorithm of Biswas and Barman [2018]
for $t \in [k]$ **do**
  Partition M'_t into groups of size n (last one may be smaller than n) as follows: $C_{t,1} \leftarrow \{g'_{t,1}, \ldots, g'_{t,n}\}$, $C_{t,2} \leftarrow \{g'_{t,n+1}, \ldots, g'_{t,2n}\}$, $\ldots$, $C_{t,\lceil |M'_t|/n \rceil} \leftarrow \{g'_{t,(\lceil |M'_t|/n \rceil - 1)n+1}, \cdots g'_{t,|M'_t|}\}$
end for
$A' \leftarrow \mathrm{BiswasBarmanCC}(\bigcup_{t \in [k]} \bigcup_{j \in [\lceil |M'_t|/n \rceil]} C_{t,j}, v')$

// Final Allocation with Daily Picking Sequences based on A'
$A \leftarrow \emptyset$
for $t \in [k]$ **do**
  for $j \in [|M_t|]$ **do**
    $i \leftarrow$ Agent allocated $g'_{t,j}$ in A'
    $g \leftarrow i$'s favourite unallocated good from M_t
    $A_i \leftarrow A_i \cup \{g\}$
  end for
end for
return A

heterogeneous additive valuations, there always exists an EF1 allocation A such that $\lfloor |C_\ell|/n \rfloor \leqslant |A_i \cap C_\ell| \leqslant \lceil |C_\ell|/n \rceil$ for every agent $i \in N$ and $\ell \in [p]$, and such an allocation can be computed in polynomial time.

Algorithm 1 invokes the algorithm of Theorem 2 on the following instance. Fix a day t. Recall that agents have identical ranked preferences for M'_t given as $g'_{t,1} \succcurlyeq \ldots \succcurlyeq g'_{t,|M'_t|}$. Divide M'_t into groups of size n in the decreasing order of value, breaking ties arbitrarily and letting the last group have possibly fewer than n goods: that is, let $C_{t,1} = \{g'_{t,1}, \ldots, g'_{t,n}\}$, $C_{t,2} = \{g'_{t,n+1}, \ldots, g'_{t,2n}\}$, and so on. By Theorem 2, we find an allocation A' that is an EF1 allocation for $\bigcup_{t \in [k]} M'_t$ and v', and that each agent is allocated at most one good from $C_{t,j}$ for all t and j.

Final Allocation. Algorithm 1 then takes the allocation A' and, for each day $M'_t = \{g'_{t,1}, \ldots, g'_{t,|M'_t|}\}$, allocates M_t through a "serial dictatorship" with the picking sequence derived from A'. First, the agent that is allocated $g'_{t,1}$ in A' will pick their most favourite good from M_t (the original set of goods for day t); next, the owner of $g'_{t,2}$ picks their most favourite good among the remaining goods of M_t; and so on. We now prove that the resulting allocation A is SD-EF1 per day and PROP1 overall.

Lemma 1. *Algorithm 1 returns an allocation that is SD-EF1 per day.*

Proof sketch. In the picking sequence for each day t, due to partitioning of the goods $C_{t,1}, \ldots, C_{t,\lceil |M'_t|/n \rceil}$ and the property that A' satisfies the "at most one per group" requirement, each agent appears exactly once in the first n positions, once in the next n positions, and so on. Additionally, each agent appears at most once among the last $|M_t| \bmod n$ positions. Such a picking order is known as "recursively-balanced", and is known to yield SD-EF1 [Aziz, 2020]. A detailed proof appears in Appendix C. □

To prove that A is PROP1 overall, we use the following technical lemma, the proof of which appears in the Appendix C.

Lemma 2. *Let V be a multiset of m real values, and $A = \{a_1, \ldots, a_k\}$ and $A' = \{a'_1, \ldots, a'_k\}$ be two subsets of V with equal size such that $a_j \geqslant a'_j$ for all $j \in [k]$. Let $g = \max\{x : x \in V \setminus A\}$ and $g' = \max\{x : x \in V \setminus A'\}$. Then, there exists a bijection z from $A \cup \{g\}$ to $A' \cup \{g'\}$ such that $x \geqslant z(x)$ for all $x \in A \cup \{g\}$.*

Lemma 3. *Algorithm 1 returns an allocation that is PROP1 overall.*

Proof. Fix an agent $i \in N$. Take a day $t \in [k]$. Rename the goods so that $A'_i \cap M'_t = \{g'_1, \ldots, g'_{|A'_i \cap M'_t|}\}$ are the goods that i is allocated in A' in a non-increasing order of v'_i. Similarly, let $A_i \cap M_t = \{g_1, \ldots, g_{|A_i \cap M_t|}\}$ be the goods i picks according to the picking sequence in order. That is, g_1 is the good picked corresponding to g'_1, g_2 corresponding to g'_2, and so on.

Towards invoking Lemma 2, a helpful observation is that $v_i(g_j) \geqslant v'_i(g'_j)$ for all $j \in [|A_i \cap M_t|]$. Suppose g'_j is the r-th preferred good among M'_t. Since M_t and M'_t share the same multiset of utility values for i, and g_j is the top pick of i when $r-1$ goods are picked, g_j is at least as good as i's r-th most preferred good among M_t (and hence, M'_t). This argument, combined across all days, implies existence of a bijection $z_i : A_i \to A'_i$ such that $v_i(g) \geqslant v'_i(z_i(g))$ for all $g \in A_i$.

By invoking Lemma 2 with $A \leftarrow A_i$ and $A' \leftarrow A'_i$ over the multiset V being i's utility values, we have that

$$v_i(A_i) + \max_{g \notin A_i} v_i(g) \geqslant v'_i(A'_i) + \max_{g \notin A'_i} v'_i(g)$$

Every EF1 allocation is also PROP1 [Conitzer *et al.*, 2017], therefore, since A' is EF1 (Theorem 2), we have that

$$v'_i(A'_i) + \max_{g \notin A'_i} v'_i(g) \geqslant \frac{1}{n} v'_i(M') = \frac{1}{n} v_i(M),$$

the last equality being true from the way we constructed M'. Combining the two inequalities above, we get $v_i(A_i) + \max_{g \notin A_i} v_i(g) \geqslant \frac{1}{n} v_i(M)$. Thus, A is PROP1. □

It is worth noting that PROP1 is not a monotonic property, i.e., if $v_i(A_i) \geqslant v_i(A'_i)$ and A'_i is PROP1, it is possible that A_i is not PROP1 (as the best good for i in $M \setminus A_i$ could be worth less than the best good in $M \setminus A'_i$). This is why we needed to use a more involved argument in Lemmas 2 and 3.

4 Two Agents

In this section, we consider temporal fair division with two agents. We provide a complete picture of temporal fairness notions that can be guaranteed in this case. We establish a strong positive result, then show that it is the best possible by producing counterexamples for stronger desiderata.

4.1 Possibilities

Our main goal in this section is to show that SD-EF1 per day and EF1 up to each day can be achieved for two agents. We begin by introducing an *envy-balancing lemma*, a powerful tool for finding temporal allocations to two agents. We later use this lemma to derive not only the aforementioned guarantee, but also other appealing guarantees.

Definition 8 (Cancelling Allocations)**.** *We say that allocations B and B' of a set of goods S to two agents* cancel out *if $v_i(B_i) + v_i(B'_i) \geqslant v_i(B_{3-i}) + v_i(B'_{3-i}), \forall i \in [2]$. In words, they cancel out if hypothetically allocating two copies of each good in S, one according to B and the other according to B', achieves (exact) envy-freeness.*

Lemma 4 (Envy-Balancing Lemma)**.** *Suppose that for each day $t \in [k]$, we are given two EF1 allocations B_t and B'_t of the set of goods M_t that cancel out. Then, we can compute, in polynomial time, an allocation A of the set of all goods $M = \cup_{t\in[k]} M_t$ that is EF1 up to each day and $A_{M_t} \in \{B_t, B'_t\}$ for each day $t \in [k]$.*

Note that the lemma achieves EF1 up to each day while not only retaining the EF1 per day property of the input allocations, but in fact by using exactly one of the two input allocations on each day. Thus, if the per day allocations given as input satisfy properties stronger than EF1, those properties are also retained per day; this is important as we will use this lemma to derive such stronger per day guarantees.

We include the proof of the lemma in Appendix D. At a high level, it works by realizing that when two allocations cancel out, we know that both agents have at least one of the two allocations where they feel no envy (they like the bundle they were given more than the bundle given to the other agent). With this in mind, we can carefully choose which allocation to assign on each day, in such a way that whenever an agent is feeling too much envy, we can give them their preferred allocation on that day, always keeping envy levels "balanced" after each time-step.

For readers familiar with the "informed" model of online fair division from He *et al.* [2019], this can be seen as a generalization of their two agent algorithm. While they achieve EF1 up to each day while allocating a single good during each time step, we provide a similar guarantee while allocating batches of goods and simultaneously maintaining fairness over the batches.

We are now ready to show that the pair of allocations required by the envy-balancing lemma — both satisfying EF1 and cancelling each other out — exists and can be computed in polynomial time. In fact, we find a single partition $(B_{t,1}, B_{t,2})$ of the goods in M_t such that both $B_t = (B_{t,1}, B_{t,2})$ and $B'_t = (B_{t,2}, B_{t,1})$ satisfy the stronger property of SD-EF1.

Lemma 5. *Given the preferences of two agents over a set of goods M_t, one can efficiently compute a partition $(B_{t,1}, B_{t,2})$ of M_t such that both $B_t = (B_{t,1}, B_{t,2})$ and $B'_t = (B_{t,2}, B_{t,1})$ are SD-EF1 allocations.*

We can plug these allocations into the envy-balancing lemma (Lemma 4) to get our desired main result.

Theorem 3. *For temporal fair division with $n = 2$ agents, an allocation that is SD-EF1 per day and EF1 up to each day exists and can be computed in polynomial time.*

We include both the proofs for Lemma 5 and Theorem 3 in Appendix D. For the interested reader, we also include some notes on the connection between Lemma 5 and the well-known Bihierarchy Theorem from Budish *et al.* [2013], as well as how to achieve other fairness notions per day while using the envy-balancing lemma, such as EFX and EF1+PO.

4.2 Impossibilities

We have shown that when there are only two agents, the strong guarantee of EF1 up to each day can be obtained along with the guarantee of SD-EF1 per day. However, one wonders if even stronger guarantees are possible, such as strengthening EF1 up to each day to SD-EF1 up to each day. We find that not only is this strengthening impossible, but even if we relax SD-EF1 up to each day to SD-EF1 overall, it is impossible to achieve alongside EF1 per day. Together, these two impossibility results prove that our guarantee of SD-EF1 per day and EF1 up to each day from Theorem 3 is the strongest possible in the hierarchy shown in Figure 1. We include the counterexamples proving both these claims in Appendix D.

Theorem 4. *For temporal fair division with $n = 2$ agents, SD-EF1 up to each day cannot be guaranteed.*

Theorem 5. *For temporal fair division with $n = 2$ agents, EF1 per day and SD-EF1 overall cannot be guaranteed simultaneously.*

4.3 Two Agents and Further Restrictions

As the final part of this section, we note that when further restrictions are placed on two-agent instances, we receive even stronger results. Particularly, in Appendix D, we prove and provide a discussion of the following theorem.

Theorem 6. *For temporal fair division with $n = 2$ agents and identical days, an allocation that is SD-EF1 per day, SD-EF1 up to each day, and SD-EF up to each even day exists and can be computed in polynomial time.*

5 Identical Orderings

We next look at instances where agents have identical orderings over all goods in M. Not only are results in this setting practically useful, as there many real life scenarios where participants agree on the ordinal ranking of goods, but results under identical orderings are also very technically useful. As can be seen from our main result in Section 3, reducing a general setting to one where agents have similar orderings over the goods can lead to fairness guarantees in the original setting. We will show in future sections that the possibility results we develop here can be applied as black-boxes to

achieve strong results in scenarios where agents have heterogeneous orderings.

Possibilities. Our main result for the case of identical orderings is the following theorem:

Theorem 7. *For temporal fair division with identical orderings, an allocation that is SD-EF1 per day and SD-EF1 overall exists and can be computed in polynomial time.*

We provide a detailed proof of Theorem 7 in the Appendix E. Intuitively, we accomplish this by creating two partitions over the set of goods M, labeled P_1 and P_2, which are defined below.

$$P_1 = \left\{ T(M_t, nr) \setminus T(M_t, n(r-1)) : r \in \left[\left\lceil \frac{|M_t|}{n} \right\rceil \right], t \in [k] \right\},$$
$$P_2 = \left\{ T(M, nr) \setminus T(M, n(r-1)) : r \in \left[\left\lceil \frac{|M|}{n} \right\rceil \right] \right\}.$$

In words, P_2 splits the entire set of goods M into the agents' most preferred n goods, their next most preferred n goods, etc. P_1 does a similar partitioning over M, but does the partitioning separately for each day. It is known that when agents have identical orderings, guaranteeing that each agent receives 1 good from their n favourite goods, 1 good from their next n favourite goods, etc. will guarantee an SD-EF1 allocation. We prove that it is possible to construct an allocation where each agent gets exactly 1 good from each set in P_1 and from each set in P_2, thereby ensuring SD-EF1 per day and overall.

To an initiated reader, the problem of finding an allocation that meets the above constraints may be immediately reminiscent of the bihierarchy framework of Budish *et al.* [2013]. We require that from each set of (at most) n goods out of a family of sets, each agent receives (at most) one good. The sets produced by each desideratum are mutually non-overlapping, forming a "hierarchy", but the sets produced by one desideratum can be overlapping with those produced by the other, resulting in two different hierarchies. However, the problem is that with $n > 2$ agents, we have a third set of constraints: each good must be assigned to (exactly) one agent. This forms a third hierarchy (which cannot be assimilated into either of the two previous hierarchies), preventing one from applying the bihierarchy framework.

Impossibilities. In the case of 2 agents, imposing the additional restriction of identical days allowed for very strong results, making it possible to satisfy SD-EF1 per day and SD-EF1 up to each day. This is unfortunately not the case when the identical days restriction is imposed in addition to identical preferences. We show via a fascinating counterexample in Appendix E that even when an instance has identical preferences and identical days, it is not always possible to even achieve SD-EF1 up to each day by itself.

Theorem 8. *For temporal fair division with identical days and identical preferences, SD-EF1 up to each day cannot be guaranteed.*

6 Identical Days

In Section 3, we showed that in the general model, we can achieve SD-EF1 per day and PROP1 overall. In this section, we will show if we assume the additional restriction that the sets of goods which arrive on each day are identical, then a slightly stronger guarantee can be achieved overall.

Theorem 9. *For any temporal fair division instance with identical days, it is possible to find an allocation that is SD-EF1 per day and SD-PROP1 overall in polynomial time.*

To achieve these guarantees, we use an algorithm that is almost identical to Algorithm 1, the algorithm which was used to achieve SD-EF1 per day and PROP1 overall in the general case, but with one major change.

Algorithm 1 first finds an EF1 overall allocation in a reduced version of the problem where agents have identical orderings over all the goods on each day, and uses that to construct an allocation in the original instance that is PROP1 overall. The key insight that can be leveraged to get stronger guarantees in this less general setting is that when we have identical days, we know that the output of the "Identical Ordering Transformation" from Algorithm 1 will result in an instance where all agents have identical orderings over the entire set of goods M, not just over the goods from each individual day. This stronger guarantee from the Identical Ordering Transformation allows us to use Theorem 7 to find an SD-EF1 per day and SD-EF1 overall allocation over the reduced instance (rather than the algorithm of Biswas and Barman [2018] which only guarantees EF1 overall). We can then use this allocation as the basis for the picking order that Algorithm 1 uses to construct the final allocation over the original instance. We include a detailed proof of Theorem 9 in Appendix F, including why an SD-EF1 allocation over the identical orderings instance will lead to a SD-PROP1 allocation in the original instance.

7 Discussion

In this work, we are able to find possibility and impossibility results that give a picture of what can be achieved in the temporal fair division model. This picture is quite clear when focusing on special cases such as two agents or identical orderings. However, we still leave many questions open for future work. All of the entries in Table 1 marked by a "?" remain open. The most interesting question in the general setting is:

> **Open Question:** In temporal fair division, does an allocation that is SD-EF1 (or EF1) per day and EF1 overall always exist?

Other interesting open questions include the existence of EF1 up to each day under identical orderings or identical days, and the existence of (SD-)EF1 per day and (SD-)EF1 overall under identical days.

Finally, it would be an interesting further direction to take a more abstract view of the temporal fair division model. In Appendix G, we introduce a generalized model of temporal fair division, where a fair allocation must be found simultaneously over a set of goods, and over each set in a collection of subsets of those goods. It would be very interesting to explore this interpretation of our model further.

Acknowledgments

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

References

Parand. A. Alamdari, Toryn Q. Klassen, Elliot Creager, and Sheila A. Mcilraith. Remembering to be fair: Non-Markovian fairness in sequential decision making. In *41th*, pages 906–920, 2024.

Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A Voudouris. Fair division of indivisible goods: A survey. In *31st*, pages 5385–5393, 2022.

Haris Aziz, Serge Gaspers, Simon Mackenzie, and Toby Walsh. Fair assignment of indivisible objects under ordinal preferences. *Artificial Intelligence*, 227:71–92, 2015.

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

Haris Aziz. Simultaneously achieving ex-ante and ex-post fairness. In *21st*, pages 341–355, 2020.

Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. Online nash social welfare maximization with predictions. In *33rd*, pages 1–19, 2022.

Siddhartha Banerjee, Vasilis Gkatzelis, Safwan Hossain, Billy Jin, Evi Micha, and Nisarg Shah. Proportionally fair online allocation of public goods with predictions. In *32nd*, pages 20–28, 2023.

Gerdus Benadè, Aleksandr M Kazachkov, Ariel D Procaccia, Alexandros Psomas, and David Zeng. Fair and efficient online allocations. *Operations Research*, 72(4):1438–1452, 2024.

Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research*, 61(1):73–87, 2013.

Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *27th*, pages 91–97, 2018.

Anna Bogomolnaia, Hervé Moulin, and Fedor Sandomirskiy. On the fair division of a random object. *Management Science*, 68(2):1174–1194, 2022.

Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. In *13th*, pages 1321–1328, 2014.

Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Designing random allocation mechanisms: Theory and applications. *American Economic Review*, 103(2):585–623, 2013.

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

Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In *18th*, pages 629–646, 2017.

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

Edith Elkind, Alexander Lam, Mohamad Latifian, Tzeh Yuan Neoh, and Nicholas Teh. Temporal fair division of indivisible items. arXiv:2410.14593, 2024.

Edith Elkind, Svetlana Obraztsova, and Nicholas Teh. Temporal fairness in multiwinner voting. In *38th*, pages 22633–22640, 2024.

Rupert Freeman, Evi Micha, and Nisarg Shah. Two-sided matching meets fair division. In *30th*, pages 203–209, 2021.

Ali Ghodsi, Matei Zaharia, Scott Shenker, and Ion Stoica. Choosy: Max-min fair sharing for datacenter jobs with constraints. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 365–378, 2013.

Vasilis Gkatzelis, Alexandros Psomas, and Xizhi Tan. Fair and efficient online allocations with normalized valuations. In *35th*, pages 5440–5447, 2021.

Sreenivas Gollapudi, Kostas Kollias, and Benjamin Plaut. Almost envy-free repeated matching in two-sided markets. In *16th*, pages 3–16, 2020.

Jiafan He, Christos-Alexandros Psomas, Ariel D Procaccia, and David Zeng. Achieving a fairer future by changing the past. In *21st*, pages 348–353, 2019.

Ayumi Igarashi, Martin Lackner, Oliviero Nardi, and Arianna Novaro. Repeated fair allocation of indivisible items. In *38th*, pages 9781–9789, 2024.

Ian Kash, Ariel D Procaccia, and Nisarg Shah. No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research*, 51:579–603, 2014.

Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. Webuildai: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):181:1–181:35, 2019.

Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *6th*, pages 125–131, 2004.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22:1–22:19, 2007.

Holmes E Miller, William P Pierskalla, and Gustave J Rath. Nurse scheduling using mathematical programming. *Operations Research*, 24(5):857–870, 1976.

Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. In *29th*, pages 2584–2603, 2018.

David Zeng and Alexandros Psomas. Fairness-efficiency tradeoffs in dynamic fair division. In *21st*, pages 911–912, 2020.

Appendix

A Additional Related Work

Repeated (or many-to-many) matching. Both our works are inspired by the earlier work of Gollapudi *et al.* [2020], who consider the repeated two-sided matching problem, where there are n agents on each side of a two-sided market with agents on each side having preferences over those on the other side, and the goal is to compute a perfect matching on each day over a period of days. They also seek guarantees such as EF1 up to each day. However, their positive results are only for binary valuations, and they leave achieving EF1 (for both sides) up to each day for general additive valuations as an open question. Finally, note that repeated perfect matching effectively produces a many-to-many matching. Freeman *et al.* [2021] study how to achieve EF1 (for both sides) in this setting, which can be viewed as an EF1 overall guarantee. They show how to achieve it when agents on each side have identical preferences, but leave it open for the case of general additive valuations. Note that unlike in fair division, EF1 overall is not straightforward in their case because EF1 needs to be achieved for agents on both sides simultaneously.

Online Fair Division In addition to the online fair division work mentioned in the main body, Benadè *et al.* [2024] show that $O(\sqrt{k \log n})$ envy can be maintained up to k days, and also point out that randomized algorithms may have much greater power against a nonadaptive adversary, who sets the full instance before the algorithm starts making random choices, with no super-constant envy lower bound known for this case. Online fair division with a nonadaptive adversary is still a stronger model than temporal fair division due to the fact that the algorithm does not have knowledge of what goods will arrive in future time periods.

Constrained fair division. We remarked that achieving an overall fairness guarantee can be reduced to the one-shot fair division model, taking an instance with the set of all goods M. When we additionally seek a per-day fairness guarantee, this can be modeled as a constraint on the space of feasible allocations, and the question becomes whether there is a *constrained allocation* that still achieves the desired fairness guarantee. This model of constrained fair division has also been studied in the literature. Biswas and Barman [2018] study a model with cardinality constraints, where M is partitioned into categories $(C_1, \ldots, C_p)$ and an allocation A is feasible only if $|A_i \cap C_\ell| \leqslant \lceil |C_\ell|/n \rceil$ for all i, ℓ. That is, the allocation should divide the goods from each category as evenly as possible (in a "balanced" manner). As we remark in Section 5, when agents have identical orderings over the goods, the SD-EF1 per day constraint can be reduced to a cardinality constraint, immediately yielding an allocation that is SD-EF1 per day and EF1 overall. However, for this case, we are able to achieve the stronger guarantee of SD-EF1 per day and SD-EF1 overall. The algorithm of Biswas and Barman [2018] relies on the envy-cycle elimination technique of Lipton *et al.* [2004] at the overall scale, which is very much reliant on the exact cardinal values, and hence, fails to achieve SD-EF1 overall. We are able to make a better algorithm, round robin, work via a non-trivial connection to the strong perfect graph theorem. Cardinality constraints (partition matroid constraints) have been generalized to matroid constraints, and the existence of an EF1 allocation subject to matroid feasibility constraints is a major open question [Biswas and Barman, 2018; Dror *et al.*, 2023]. For further discussion, see Section 7. Finally, the bihierarchy framework of Budish *et al.* [2013] can also be viewed as a method for finding a constrained allocation, which we use in some of our results. Although, our most interesting results deal with sets of constraints that go beyond bihierarchies.

B Missing Proofs from Section 2

B.1 Properties of SD-EF1 Allocations

We will begin by noting some useful properties of SD-EF1 allocations. Specifically, in Section 2, we establish the function $T_i(S, r)$, which returns agent i's top r ordered goods from a set S, breaking ties according to some arbitrary rule consistent across all agents. In contrast, the definitions for SD-EF1 and SD-PROP1 are based around sets of the form $H_i(S, g) = \{g' \in S : g' \succcurlyeq_i g\}$, which returns all goods from S that are weakly preferred to g. As will be seen in other proofs in this Appendix, it is often very useful to be able to look at a set of exactly size r of some agent's top goods, which the sets $H_i(S, g)$ do not allow for. Below, we will show relations between the $T_i(S, r)$ function, and the SD-EF1 definition, that allows us to often use it without loss of generality when proving statements about SD-EF1 and SD-PROP1, vastly simplifying many of our proofs.

Observation 1. *For any agent $i \in N$ and good $g \in S$, if $|H_i(S, g)| = r$, then $T_i(S, r) = H_i(S, g)$, regardless of the arbitrary tie-breaking order dictated by T_i.*

Proof. This follows from the fact that agent i's ordering $\succcurlyeq_i$ over the goods in S will be transitive. $|H_i(S, g)| = r$ means that there are *exactly* r goods that i weakly prefers to g. It must be the case that for any good $g^+ \in H_i(S, g)$, $g^- \in S \setminus H_i(S, g)$, we must have that $g^+ \succ_i g^-$. If there were some goods $g^+ \in H_i(S, g)$, $g^- \in S \setminus H_i(S, g)$ such that $g^- \succcurlyeq_i g^+$, then by the transitivity of i's ordering, we know that $g^- \succcurlyeq_i g^+ \succcurlyeq_i g$ is true, contradicting that fact that $g^- \in S \setminus H_i(S, g)$.

Therefore, when $|H_i(S, g)| = r$, we know that there are r goods in S that agent i strictly prefers to all other goods in S. It is clear that $T_i(S, r)$ will return exactly those goods, and will not need to use its tie-breaking order in this case. □

With this observation in mind, we can now list several necessary and sufficient conditions for an allocation to be SD-EF1, which relate it directly to the T_i function.

Proposition 1. *Let A be an allocation of a set of goods S.*

- *(Sufficiency) If $|A_i \cap T_i(S,r)| \geqslant |A_j \cap T_i(S,r)| - 1$ for all $i \in N$ and $r \in [|S|]$, then A is SD-EF1. If $n = 2$, the condition can be written as $|A_i \cap T_i(S,r)| \geqslant \lfloor r/n \rfloor$. If $T_i(S,r) = T_j(S,r)$ for all $i,j \in N$, the condition can be written as $|A_i \cap T_i(S,r)| \in \{\lfloor r/n \rfloor, \lceil r/n \rceil\}$.*
- *(Necessity) If A is SD-EF1, then $|A_i \cap T_i(S,r)| \geqslant \lfloor r/n \rfloor$ for all $i \in N$ and $r \in [|S|]$ conditioned on $g \succ_i g'$ for all $g \in T_i(S,r)$ and $g' \in S \setminus T_i(S,r)$. Further, if $T_i(S,r) = T_j(S,r)$ for all $i,j \in N$, then the condition can be written as $|A_i \cap T_i(S,r)| \in \{\lfloor r/n \rfloor, \lceil r/n \rceil\}$.*

Proof. Below are the proofs for both the sufficiency and necessity conditions:

- (Sufficiency)

  General Case Assume that some agent i has the ordering $g_1 \succcurlyeq_i g_2 \ldots \succcurlyeq_i g_s$ over the set of goods S, where some preferences may be strict. For contradiction, assume that for all $r \in [|S|]$ and $j \in N$, $|A_i \cap T_i(S,r)| \geqslant |A_j \cap T_i(S,r)| - 1$, but for some $g \in S$ and $j \in N$, we have $|A_i \cap H_i(S,g)| < |(A_j \setminus \{g_j\}) \cap H_i(S,g)|$, where g_j is agent i's most preferred good from A_j.

  Let $|H_i(S,g)| = r'$. Then from Observation 1, we know that $T_i(S,r') = H_i(S,g)$, and from our assumption, we know that $|A_i \cap T_i(S,r')| \geqslant |A_j \cap T_i(S,r')| - 1$. Note that this is equivalent to saying that $|A_i \cap H_i(S,g)| \geqslant |A_j \cap H_i(S,g)| - 1$.

  To get the contradiction, we just need to notice that $|(A_j \setminus \{g^*\}) \cap H_i(S,g)| \geqslant |A_j \cap H_i(S,g)| - 1$ for all $g^* \in A_j$, since if $g^* \in H_i(S,g)$, then we have $|(A_j \setminus \{g^*\}) \cap H_i(S,g)| = |A_j \cap H_i(S,g)| - 1$, and otherwise we have $|(A_j \setminus \{g^*\}) \cap H_i(S,g)| = |A_j \cap H_i(S,g)|$.

  2 Agents For the case of 2 agents (Agent i and Agent j), it is sufficient to notice that $|A_i \cap T_i(S,r)| \geqslant \lfloor r/2 \rfloor$ implies that $|A_j \cap T_i(S,r)| \leqslant \lceil r/2 \rceil$, since $A_j = S \setminus A_i$ when there are two agents. Since $\lfloor r/2 \rfloor \geqslant \lceil r/2 \rceil - 1$, this gives us that $|A_i \cap T_i(S,r)| \geqslant |A_j \cap T_i(S,r)| - 1$ for all $i,j \in N$ and $r \in [|S|]$, which we know implies SD-EF1.

  Identical Orderings Finally, in this case, we can see that for all $i \in N$, $r \in [|S|]$, $|A_i \cap T_i(S,r)| \in \{\lfloor r/n \rfloor, \lceil r/n \rceil\}$ implies that $|A_i \cap T_i(S,r)| \geqslant |A_j \cap T_i(S,r)| - 1$. This is due to the fact that since $T_i(S,r) = T_j(S,r)$ for all i,j, we must have that $|A_i \cap T_i(S,r)| \geqslant \lfloor r/n \rfloor$ and $|A_j \cap T_i(S,r)| \leqslant \lceil r/n \rceil$.

- (Necessity)

  General Case For contradiction, assume this is false. There is some allocation A over a set of goods S, some agent i and some $r \in [|S|]$ such that A is SD-EF1 and $|A_i \cap T_i(S,r)| < \lfloor r/n \rfloor$, and for all goods $g \in T_i(S,r)$, $g' \in S \setminus T_i(S,r)$, $g \succ_i g'$. Assume that agent i has the following order over the goods in S, $g_1 \succcurlyeq_i g_2 \ldots \succcurlyeq_i g_s$ where some of the preferences may be strict.

  Let g^* be the good in $T_i(S,r)$ that is not strictly preferred to any other good in $T_i(S,r)$. Since agent i strictly prefers all goods in $T_i(S,r)$ to all goods that are not, we must have that $|H_i(S,g^*)| = r$, and thus by Observation 1, $T_i(S,r) = H_i(S,g^*)$. Therefore, we have that $|A_i \cap H_i(S,g^*)| < \lfloor r/n \rfloor$.

  When $|A_i \cap H_i(S,g^*)| < \lfloor r/n \rfloor$, note that by the fact that every good must be allocated to one of the n agents, there must exist some $j \in N$ such that $|A_j \cap H_i(S,g^*)| \geqslant \lfloor r/n \rfloor + 1$. This means that for this j, we have $|A_i \cap H_i(S,g^*)| < |A_j \cap H_i(S,g^*)| - 1$. But that means that for any $g_j \in A_j$, we must have $|A_i \cap H_i(S,g^*)| < |(A_j \setminus \{g_j\}) \cap H_i(S,g^*)|$, contradicting the fact that A is SD-EF1.

  Identical Orderings In the case where agents have identical orderings over the goods, it can be seen that $|A_i \cap T_i(S,r)| \in \{\lfloor r/n \rfloor, \lceil r/n \rceil\}$ follows as a consequence of the necessity statement in the general case. When every agent gets at least items $\lfloor r/n \rfloor$ of some set of size r, there must be some agent i who gets exactly $\lfloor r/n \rfloor$ items from the set. When n divides r, then each agent getting at least $\lfloor r/n \rfloor$ goods implies that each agent gets exactly $r/n = \lceil r/n \rceil$ goods. Otherwise, we will have $\lfloor r/n \rfloor = \lceil r/n \rceil - 1$, and therefore, there can be no other agent j such that $|A_j \cap T(S,r)| > \lceil r/n \rceil$, or else it would immediately follow that $|A_i \cap H_i(S,g^*)| < |(A_j \setminus \{g_j\}) \cap H_i(S,g^*)|$ for some g^*, and all $g_j \in A_j$.

□

B.2 Properties of SD-PROP1

Here we formalize the fact that SD-EF1 implies SD-PROP1, and that SD-PROP1 implies PROP1, which completes the hereditary relationships of desiderata shown in Figure 1.

Proposition 2. *If an allocation A over a set of goods S is SD-EF1, then it will also be SD-PROP1.*

Proof. Assume that this is false, and for some allocation A over a set of goods S, we have that A is SD-EF1, but for some $i \in N$ and some $g^* \in S$, we have that $|A_i \cap H_i(M, g^*)| < \lfloor {}^{|H_i(S,g^*)|}/_{n} \rfloor$.

Note that since every good must be allocated to some agent, if $|A_i \cap H_i(S, g^*)| < \lfloor {}^{|H_i(S,g^*)|}/_{n} \rfloor$ were true, then there must be some other agent $j \in N$ such that $|A_j \cap H_i(S, g^*)| \geqslant \lfloor {}^{|H_i(S,g^*)|}/_{n} \rfloor + 1$. This would directly imply that $|A_i \cap H_i(S, g^*)| < |A_j \cap H_i(S, g^*)| - 1$, so clearly $|A_i \cap H_i(S, g^*)| < |(A_j \setminus \{g_j\}) \cap H_i(S, g^*)|$ for any $g_j \in A_j$. This contradicts the fact that A is SD-EF1. □

Proposition 3. *If an allocation A over a set of goods S is SD-PROP1, then it will also be PROP1.*

Proof. Assume that some agent i has the ordering $g_1 \succcurlyeq_i g_2 \ldots \succcurlyeq_i g_{|S|}$ over the set of goods S, where some preferences may be strict, and ties are broken based on the the tie-breaking ordering of the function T_i.

Define the bundle $P_i = \{g_1, g_{n+1}, g_{2n+1}, \ldots\}$. It must be the case that $v_i(P_i) \geqslant \frac{1}{n} v_i(S)$. To see this, partition S into disjoint subsets of size n in the form of $C_1 = \{g_1, \ldots, g_n\}, C_2 = \{g_{n+1}, g_{2n}\}, C_3 = \{g_{2n+1}, g_{3n}\}$, and so on. Notice that in each subset C, there is a single good in $g \in P_i \cap C$, and that good is weakly preferred to all other goods in C. Thus for each C, we must have that $v_i(g) \geqslant \frac{1}{n} v_i(C)$. Summing over all subsets we get our desired inequality.

Next, notice that for any $g \in S$, $|A_i \cap H_i(S, g)| \geqslant \lfloor {}^{|H_i(S,g)|}/_{n} \rfloor$ implies that $|(A_i \cup \{g^*\}) \cap H_i(M, g)| \geqslant \lceil {}^{|H_i(S,g)|}/_{n} \rceil$, where g^* is agent i's most preferred good in $S \setminus A_i$. This is because if $|A_i \cap H_i(S, g)| \leqslant \lceil {}^{|H_i(S,g)|}/_{n} \rceil$ were true, then there would need to be some good g' such that $g' \in H_i(S, g)$ and $g' \notin A_i$. Since g^* is i's most preferred good that is not in A_i, then we know that $g^* \succcurlyeq_i g'$, and thus $g^* \in H_i(S, g)$.

To complete the proof, we will show that $A_i \cup \{g^*\} \succcurlyeq_i^{\text{SD}} P_i$, where g^* is agent i's most preferred good from $S \setminus A_i$. For contradiction, assume this were false, and that for some $g \in S$, $|(A_i \cup \{g^*\}) \cap H_i(S, g)| < \{P_i \cap H_i(S, g)\}$. Let $|H_i(S, g)| = r$. By Observation 1, we must have that $H_i(S, g) = T_i(S, r) = \{g_1, \ldots, g_r\}$. Because of the way that we constructed P_i, we know that there cannot be more than $\lceil {}^{r}/_{n} \rceil = \lceil {}^{|H_i(S,g)|}/_{n} \rceil$ goods from P_i in $\{g_1, \ldots, g_r\}$. However, since we know that A is SD-PROP1, it must be the case that $|(A_i \cup \{g^*\}) \cap H_i(S, g)| \geqslant \lceil {}^{|H_i(S,g)|}/_{n} \rceil$, giving a contradiction, and showing that $A_i \cup \{g^*\} \succcurlyeq_i^{\text{SD}} P_i$.

This tells us that $v_i(A_i \cup \{g^*\}) \geqslant v_i(P_i) \geqslant \frac{1}{n} v_i(S)$ for any $i \in N$. □

Note that since SD-PROP1 only considers agents' orderings over the goods in M, the above proposition implies that an allocation that is SD-PROP1 for a set of orderings, is guaranteed to be PROP1 on any set of valuation function that induce those orderings.

C Missing Proofs from Section 3

C.1 Proof of Lemma 1

Proof of Lemma 1. Let A be the allocation returned by Algorithm 1. Due to Proposition 1, to prove that A is SD-EF1 per-day, it is sufficient to show that $|A_{M_t,i} \cap T_i(M_t, r)| \geqslant |A_{M_t,j} \cap T_i(M_t, r)| - 1$ for all $i, j \in N$, $t \in [k]$, and $r \in [|M_t|]$.

Let A' represent the allocation that the algorithm finds for the identical orderings transformation instance. In each day $t \in [k]$, we have that M'_t is partitioned into sets $C_{t,1} = \{g'_{t,1}, \ldots, g'_{t,n}\}, C_{t,2} = \{g'_{t,n+1}, \ldots, g'_{t,2n}\}$, and so on, in order of utility value. Without loss of generality, assume that the ordering of the labeling of the goods is also consistent with the tie-break ordering of T_i. The allocation $A'_{M'_t}$ will guarantee that no agent receives more than 1 good from each set $C_{t,l}$ for all l. Note that this will imply that for all $i \in N$ and $r \in [|M'_t|]$, we will have $|A'_i \cap T(M'_t, r)| \in \{\lfloor r/n \rfloor, \lceil r/n \rceil\}$. By the sufficiency condition of Proposition 1, this means that $A'_{M'_t}$ will be SD-EF1.

Next, Algorithm 1 uses a picking order procedure to construct the final allocation A_{M_t} from $A'_{M'_t}$. It can be seen that each agent is assigned exactly one "pick" over the goods in M_t for each good they received in the allocation $A'_{M'_t}$, with the ordering of these picks corresponding to the ordering of M'_t induced by the T_i function. Therefore, we know that for any $r \in [|M_t|]$, after the rth pick of the procedure, each agent will have received either $\lfloor r/n \rfloor$ or $\lceil r/n \rceil$ picks. It can also be seen that after the rth pick of the procedure, $|A_i \cap T_i(M_t, r)| \geqslant \lfloor r/n \rfloor$ will be true for each agent i. This is because at each pick r, the picking agent will select their most preferred good from M_t that has not yet been picked. After pick r, only r goods from M_t have been assigned, and $|T_i(M_t, r)| = r$, so each of an agent's picks up to and including the rth overall pick of the procedure will all have been used to select an item from their top r goods from M_t.

For contradiction, assume our original claim is not true, the allocation on some day is not SD-EF1, and therefore, there exists some agents $i, j \in N$, some day $t \in [k]$, and some $r \in [|M_t|]$, such that $|A_{M_t,i} \cap T_i(M_t, r)| < |A_{M_t,j} \cap T_i(M_t, r)| - 1$.

Let r' be the last pick in the picking sequence where agent j picked a good from $T_i(M_t, r)$. We know that $|A'_i \cap T(M'_t, r')| \in \{\lfloor r'/n \rfloor, \lceil r'/n \rceil\}$, so agent i must have received at least $\lfloor r'/n \rfloor$ picks prior to pick r', and each of those picks must have been used to select a good from $T_i(M_t, r)$ (Agent i would never have used one of these picks to select a good not from $T_i(M_t, r)$, since we know there was at least one good from $T_i(M_t, r)$ available, the good that agent j selected with pick r'). Similarly, we know that $\left|A'_j \cap T(M'_t, r')\right| \in \{\lfloor r'/n \rfloor, \lceil r'/n \rceil\}$, so agent j could only have had a maximum of $\lceil r'/n \rceil$ picks up to and including the r'th overall pick. Since r' is the last pick where agent j selected a good from $T_i(M_t, r)$, that means A_j can only contain at most $\lceil r'/n \rceil$ goods from $T_i(M_t, r)$. This gives us a contradiction since $\lfloor r'/n \rfloor \geqslant \lceil r'/n \rceil - 1$. □

Algorithm 2 Envy-Balancing Algorithm

Input for $t \in [k]$, a pair of allocations (B_t^1, B_t^2) of the set of goods M_t that cancel out, with labels assigned in such a way that if neither of the allocations is Envy-Free, then $v_1(B_{t,1}^1) - v_1(B_{t,2}^1) \geqslant 0$ and $v_2(B_{t,2}^2) - v_2(B_{t,1}^2) \geqslant 0$.

Output An allocation A of the set of goods $M = \cup_{t \in [k]} M_t$

```
F ← ∅, S ← ∅, e_1 ← 0, e_2 ← 0
for t ∈ [k] do
  if v_1(B^1_{t,1}) ⩾ v_1(B^1_{t,2}) ∧ v_2(B^1_{t,2}) ⩾ v_2(B^1_{t,1}) then
    F ← F ∪ {B_t}
  else if v_1(B^2_{t,1}) ⩾ v_1(B^2_{t,2}) ∧ v_2(B^2_{t,2}) ⩾ v_2(B^2_{t,1}) then
    F ← F ∪ {B'_t}
  else
    if e_1 ⩽ 0 then
      S ← S ∪ {B^1_t}
      e_1 ← e_1 + (v_1(B^1_{t,1}) − v_1(B^1_{t,2}))
      e_2 ← e_2 + (v_2(B^1_{t,2}) − v_2(B^1_{t,1}))
    else
      S ← S ∪ {B^2_t}
      e_1 ← e_1 + (v_1(B^2_{t,1}) − v_1(B^2_{t,2}))
      e_2 ← e_2 + (v_2(B^2_{t,2}) − v_2(B^2_{t,1}))
    end if
  end if
  if e_1 ⩾ 0 ∧ e_2 ⩾ 0 then
    F ← F ∪ S
    S ← ∅, e_1 ← 0, e_2 ← 0
  else if e_1 ⩽ 0 ∧ e_2 ⩽ 0 then
    F ← F ∪ SWAP(S)
    S ← ∅, e_1 ← 0, e_2 ← 0
  end if
end for
F ← F ∪ S
A ← allocation in which M_t is allocated according to the allocation of M_t in F, for each t ∈ [k]
return A
```

C.2 Proof of Lemma 2

Proof of Lemma 2. In the case where $g \geqslant g'$, then z can be constructed by simply mapping g to g' and a_j to a'_j for all $j \in [k]$.

Now, the case where $g < g'$. Suppose g is the rth element of V (ties broken such that g is ranked lowest possible). Since g is the maximum among $V \setminus A$, the first $r-1$ elements of V are should be in A. Therefore, $\{a_1, \ldots, a_{r-1}\} \cup \{g\}$ are the top r elements of V, and we can simply create a bijection from those elements to $\{a'_1, \ldots a'_{r-1}\} \cup \{g'\}$ as desired. For the remaining $k-r$ bottom ranks of both A and A', we simply use $z(a_j) = a'_j$ since we know from the statement that $a_j \geqslant a'_j$, which completes the proof. □

D Missing Proofs from Section 4

D.1 Proof of Lemma 4

Proof of Lemma 4. For any allocation A, we will use the following language to describe the agents' relative valuations of their bundles compared to the other agent:

- For any allocation A where $v_i(A_{3-i}) > v_i(A_i)$ for some agent $i \in \{1, 2\}$, the negative value $v_i(A_i) - v_i(A_{3-i})$ will be refereed to as the "Envy" felt by Agent i in A.
- Similarly, for any allocation A where $v_i(A_i) > v_i(A_{3-i})$ for some agent $i \in \{1, 2\}$, the positive value $v_i(A_i) - v_i(A_{3-i})$ will be referred to as the "surplus utility" that the respective agents feels in A.

For each $t \in [k]$, let B_t^1, B_t^2 be two allocations over M_t that are both EF1 and cancel out. Since these allocations cancel out, we can make the following assumption without loss of generality:

For every day t, If neither of B_t^1 or B_t^2 are envy-free, then without loss of generality, we say that

$$v_1(B_{t,1}^1) > v_1(B_{t,2}^1)$$

$$v_2(B^1_{t,1}) > v_2(B^1_{t,2})$$

$$v_1(B^2_{t,1}) < v_1(B^2_{t,2})$$

$$v_2(B^2_{t,1}) < v_1(B^2_{t,2})$$

This is due to the fact that the allocations canceling out allows us to know that exactly one agent feels envy in each allocation (if they both felt envy in some allocation then the other allocation would have to be envy free in order to cancel out), and the same agent cannot feel envy in both allocations (or else their envy would clearly not cancel out). For simplicity we assume that in this case, Agent 2 always feels envy in B^1_t, and Agent 1 always feels envy in B^2_t. B^1_t can be thought of as Agent 1's "preferred" allocation, and B^2_t as Agent 2's "preferred" allocation. We can further conclude that the equalities in each agent's preferred allocation must be strict, since if any agent was indifferent between the bundles of one allocation, they could not feel any envy in the other allocation, so therefore one of the two allocations would need to be envy-free.

Algorithm 1 functions by examining each day in order, and picking one allocation from each day's pair. When the algorithm selects an allocation for some day, it puts it into one of two sets. F is the "Final" set. If an allocation is put into F, that means that it will be in the final allocation returned by the Algorithm. S is the "Swap" set. If an allocation is put into S, that means that it may be changed at some point in the future. Specifically, the algorithm may perform a SWAP on S. This means that for every day $t \in [k]$, if an allocation from day t is in S, that allocation will be replaced with the other allocation from day t that is not in S. We will refer to S_t and F_t as the contents of the sets S and F directly after iteration t of the algorithm has completed, and will refer to SWAP(S) as the contents of S if a SWAP were performed on it. With slight abuse of notation, for any set T containing allocations over some days, we will refer to A_T as the allocation induced by combining all the per-day allocations in T.

Clearly Algorithm 1 will produce an allocation that is EF1 Per-Day. We will show that it also produces an allocation that is EF1 up to each day.

To do this, we will first note that for both agents $i \in \{1, 2\}$, for any possible set S during the runtime of the algorithm, if an agent feels envy in the allocation A_S, then they will not feel envy in the allocation $A_{\text{SWAP}(S)}$. This follows from the fact that performing a SWAP on S involves replacing each allocation in $B \in S$ with an allocation that cancels out B. let $D \subseteq [k]$ be a set of days such that S contains an allocation B_t for each day $t \in D$. Let D^+ be the days where the allocation $B_t \in S$ is agent i's preferred allocation, and $D^- = D \setminus D^+$. From the fact that we know the pair of allocations (B^i_t, B^{3-i}_t) on each day cancels out, we have:

$$\sum_{t \in D^+} (v_i(B^i_{t,i}) - v_i(B^i_{t,3-i})) \geqslant \sum_{t \in D^+} (v_i(B^{3-i}_{t,3-i}) - v_i(B^{3-i}_{t,i}))$$

$$\sum_{t \in D^-} (v_i(B^{3-i}_{t,3-i}) - v_i(B^{3-i}_{t,i})) \leqslant \sum_{t \in D^-} (v_i(B^i_{t,i}) - v_i(B^i_{t,3-i}))$$

together gives us the following implication:

$$v_i(A_{S,i}) - v_i(A_{S,3-i}) = \sum_{t \in D^+} (v_i(B^i_{t,i}) - v_i(B^i_{t,3-i})) - \sum_{t \in D^-} (v_i(B^{3-i}_{t,3-i}) - v_i(B^{3-i}_{t,i})) \leqslant 0$$

implies

$$v_i(A_{\text{SWAP}(S),i}) - v_i(A_{\text{SWAP}(S),3-i}) = \sum_{t \in D^-} (v_i(B^i_{t,i}) - v_i(B^i_{t,3-1i})) - \sum_{t \in D^+} (v_i(B^{3-i}_{t,3-i}) - v_i(B^{3-i}_{t,i})) \geqslant 0$$

Along with the above facts, proving the following inductive hypothesis will be sufficient to show that the allocation returned by Algorithm 1 will be EF1 up to each day:

For all $t \in [k]$, if the following conditions hold after the $(t-1)$th iteration of the algorithm, they will hold at the tth iteration.

- A_{F_t} is an envy-free allocation.
- A_{S_t} is an EF1 allocation.
- $A_{\text{SWAP}(S_t)}$ is an EF1 allocation.

We will show this by analyzing each possible state the algorithm can be in after some iteration t.

First, we will show that this holds for some obvious cases.

- **Day t has an EF Allocation** When the pair (B^1_t, B^2_t) for some day t contains an EF allocation, then the algorithm simply adds that allocation to F. This clearly maintains the envy-freeness of F_t, and the contents of S_t will be the same as S_{t-1}.

- **During iteration t, some allocation B_t^i is added to S that causes neither agent to feel envy over $A_{S_{t-1} \cup \{B_t^i\}}$** Directly after B_t^i has been added to S, A_S will be an Envy-Free allocation, and the algorithm will move the entire current contents of S to F. A_{F_t} will remain EF since the algorithm is adding an EF allocation to it, and $A_{S_t}, A_{\text{SWAP}(S_t)}$ will trivially meet their conditions since S_t will be empty.

- **During iteration t, an allocation B_t^i is added to S that causes both agents to feel envy over $A_{S_{t-1} \cup \{B_t^i\}}$** Similarly to above, directly after B_t^i has been added to S, both agents will either feel envy in A_S, or will be indifferent between the two bundles in A_S. In this case, the algorithm will perform a SWAP on S. The allocation induced by this newly swapped S will be Envy-Free. The algorithm will then move the contents of S to F. A_{F_t}, A_{S_t}, and $A_{\text{SWAP}(S_t)}$ will meet the required conditions for the same reasons as in the case above.

- $S_{t-1} = \emptyset$ Finally, in the case where S is empty at the beginning of iteration t, and there is no EF allocation over M_t, the algorithm will add the allocation B_t^1 to S. A_{F_t} will be EF since it was not altered, A_{S_t} will be EF1 since B_t^1 is EF1, and SWAP(A_{S_t}) will be EF1 since B_t^2 is EF1.

From this, whenever the algorithm executes iteration t and was not in one of the above cases, we can conclude the following:

- Neither of B_t^1 or B_t^2 are envy-free.
- S_{t-1} was not empty. Since we know that during iteration $t-1$, if an allocation is added to S that makes both agent feel envy or both agents have surplus utility, then S_{t-1} would be empty, so we can also conclude that exactly one agent must feel envy in $A_{S_{t-1}}$, and the other must feel surplus utility (notably, neither agent can be indifferent between the bundles).
- S_t will not be empty. From this, we can conclude that exactly one agent must feel envy in A_{S_t}, while the other feels surplus utility. It also allows us to conclude that a SWAP was not performed in iteration t, as a SWAP is always proceeded with moving the contents of S to T.

We can show that in this case as well, the inductive step holds.

In this case, there will be one agent $i \in \{1, 2\}$ who feels envy in $A_{S_{t-1}}$, while the other agent feels surplus utility. During iteration t, the algorithm will select Agent i's preferred allocation B_t^i, and add it to S. By our hypothesis, we have that $A_{S_{t-1}}$ is EF1, so there must be some good $g \in A_{S_{t-1},3-i}$ that can be taken away to eliminate all Agent i's envy. Because we add in Agent i's preferred allocation from day t, we know that $v_i(A_{S_t,i}) - v_i(A_{S_t,3-i}) > v_1(A_{S_{t-1},i}) - v_1(A_{S_{t-1},3-i})$, meaning that A_{S_t} must still be EF1 with respect to Agent i, as we can still remove g from $A_{S_t,3-i}$ to eliminate all envy. We can also show that A_{S_t} will be EF1 with respect to Agent $3-i$. Since B_t^i is Agent $3-i$'s unpreferred allocation, we have that $v_{3-i}(A_{S_t,3-i}) - v_{3-i}(A_{S_t,i}) < v_{3-i}(A_{S_{t-1},3-i}) - v_{3-i}(A_{S_{t-1},i})$. However, since we know that B_t^i is EF1, we know that there exists some $g \in B_{t,i}^i$ such that $v_{3-i}(B_{t,3-i}) \geqslant v_{3-i}(B_{t,i}) - v_{3-i}(g)$. Combining this with the fact that $v_{3-i}(A_{S_{t-1},3-i}) > v_{3-i}(A_{S_{t-1},i})$, we get that $v_{3-i}(A_{S_{t-1},3-i}) + v_{3-i}(B_{t,3-i}) > v_{3-i}(A_{S_{t-1},i}) + v_{3-i}(B_{t,i}) - v_{3-i}(g)$. Since we know that $g \in A_{S_t,i}$, this gives us EF1 as desired.

The proof that $A_{\text{SWAP}(S_t)}$ can be done similarly. Notice that SWAP(S_t) will be equal to SWAP$(S_{t-1}) \cup \{B_t^{3-i}\}$. We know $A_{\text{SWAP}(S_{t-1})}$ will be EF1, and in $A_{\text{SWAP}(S_{t-1})}$, we know that agent i will feel surplus utility. Agent $3-i$ may feel envy in SWAP(S_{t-1}), but since B_t^{3-i} is their preferred bundle, they cannot be the reason why $A_{\text{SWAP}}(S_t)$ is not EF1, as the envy they feel in $A_{\text{SWAP}(S_{t-1})}$ strictly decreases from the envy they feel in $A_{\text{SWAP}(S_t)} = A_{\text{SWAP}(S_{t-1}) \cup \{B_t^{3-i}\}}$. Agent i cannot be the reason why $A_{\text{SWAP}}(S_t)$ is not EF1, since there must be some good $g \in B_t^{3-1}$ that will eliminate all of agent i's envy over $A_{\text{SWAP}(S_t)}$ when removed.

We also know that A_{F_t} will be EF since we have $A_{F_t} = A_{F_{t-1}}$. In the base case, both S and F will be empty at the beginning of the algorithm, thus all the conditions in the inductive statement will trivially be true.

Finally, note that this inductive statement being true implies that the final allocation will be EF1 up to each day. This is due to the fact that in the final allocation A outputted by the algorithm, for any $t \in [k]$, it must be true that the allocation $A_{\overline{M}_t} \in \{A_{F_t \cup S_t}, A_{F_t \cup \text{SWAP}(S_t)}\}$. This is due to the fact that the only operation that can be performed on S is a SWAP, and the algorithm only ever moves the entirety of S into F, never just part of it. Due to the fact that we know A_{F_t} will be EF and A_{S_t} and $A_{\text{SWAP}(S_t)}$ must be EF1, we know that $A_{\overline{M}_t}$ must be EF1 as well. □

D.2 Proof of Lemma 5

Proof of Lemma 5. Draw a graph $G = (M_t, E)$ with the goods in M_t as the nodes. For each agent $i \in [2]$ and $r \in [\lfloor \frac{|M_t|}{2} \rfloor]$, draw an edge between the two goods in $T_i(M_t, 2r) \setminus T_i(M_t, 2(r-1))$. The edges added for each agent form a matching, so this graph is a union of two matchings, and hence, a bipartite graph. Thus, it admits a 2-coloring $c : M_t \to \{1, 2\}$, which can be computed efficiently. Define $B_t = (B_{t,1}, B_{t,2})$, where $B_{t,i} = \{g \in M_t : c(g) = i\}$ for each $i \in [2]$.

Note that due to the way we added the edges, each agent $i \in [2]$ receives exactly r of her $2r$ most favorite goods, for each $r \in [\lfloor \frac{|M_t|}{2} \rfloor]$. This meets the sufficiency condition from Proposition 1, implying that B_t is SD-EF1.

It is easy to see that the same reasoning also shows $B'_t = (B_{t,2}, B_{t,1})$ is also SD-EF1. □

For readers familiar with the bihierarchy matrix decomposition theorem of Budish *et al.* [2013], it is worth remarking that Lemma 5 can also be derived as a corollary. Specifically, we can define a binary variable $x_g \in \{0, 1\}$ to indicate whether good g should be allocated to agent 1 (with $1 - x_g$ denoting whether it should be allocated to agent 2, and write the set of constraints:

$$\forall r \in \left[\left\lfloor \frac{|M_t|}{2} \right\rfloor\right] : \sum_{g \in T_1(M_t, 2r) \setminus T_1(M_t, 2(r-1))} x_g = 1,$$
$$\forall r \in \left[\left\lfloor \frac{|M_t|}{2} \right\rfloor\right] : \sum_{g \in T_2(M_t, 2r) \setminus T_2(M_t, 2(r-1))} (1 - x_g) = 1.$$

It is easy to notice that this constraint set forms a "bihierarchy", and since it admits a fractional solution ($x_g = 1/2$ for all g), the result of Budish *et al.* [2013] implies the existence and polynomial-time computability of an integral allocation satisfying them, which is what we need. However, we provide a more direct proof for our specific constraint set because it is simpler to understand and leads to a faster algorithm.

D.3 Proof of Theorem 3

Proof of Theorem 3. Consider the allocations B_t and B'_t generated by Lemma 5. Each is SD-EF1 and because they use the same partition of M_t into bundles but do the opposite assignments, they trivially cancel out. Thus, due to Lemma 4, these can be combined to compute an allocation that is SD-EF1 per day and EF1 up to each day. □

D.4 Other Fairness Desiderata that can be guaranteed by the Envy-Balancing Lemma

In the main body of the paper, we remarked that if one can always find two allocations over each day M_t that cancel out, and satisfy some fairness desiderata that implies EF1, then one can find an allocation over the entire set of goods that satisfies that desiderata per day, along with EF1 up to each day. We showed that it is always possible to find such a pair of allocations that satisfies SD-EF1. Below, we will show the same for two other interesting strengthenings of EF1.

Definition 9 (Envy-Freeness Up to Any Good (EFX))**.** *An allocation A of a set of goods S is* envy-free up to any good *(EFX) if for all $i, j \in N$ and $g \in A_j$, $v_i(A_i) \geqslant v_i(A_j \setminus \{g\})$.*

Theorem 10. *For temporal fair division with $n = 2$ agents, an allocation that is EFX per day and EF1 up to each day exists.*

Proof. Due to Lemma 4, it is sufficient to prove that for any day t, there exist EFX allocations B_t and B'_t of the goods in M_t that cancel out.

We use the CUTANDCHOOSE++algorithm of Plaut and Roughgarden [2018, Algorithm 4.2] to produce the required two EFX allocations and show that they cancel out. Because their algorithm has a simpler description for additive valuations (which we focus on), we explicitly describe the construction here. Allocation B_t is constructed as follows.

1. Find a partition (P, Q) of M_t that minimizes $|v_1(P) - v_1(Q)|$. With loss of generality, assume that $v_1(P) \geqslant v_1(Q)$.
2. If there are any goods $g \in P$ such that $v_1(g) = 0$, move them to Q. Note that this does not change $v_1(P)$ or $v_1(Q)$.
3. Allow agent 2 to pick their preferred bundle, and assign the other bundle to agent 1.

Allocation B'_t is computed similarly, but reversing the roles of agents 1 (who picks) and agent 2 (who cuts).

Plaut and Roughgarden [2018, Theorem 4.3] show that the resulting allocations, B_t and B'_t are EFX. It remains to show that they cancel out. Due to symmetry, we simply need to argue the cancellation for agent 1, i.e.,

$$v_1(B_{t,1}) + v_1(B'_{t,1}) \geqslant v_1(B_{t,2}) + v_1(B'_{t,2}) \Leftrightarrow v_1(B'_{t,1}) - v_1(B'_{t,2}) \geqslant v_1(B_{t,2}) - v_1(B_{t,1}).$$

Note that $v_1(B'_{t,1}) - v_1(B'_{t,2}) \geqslant 0$ because agent 1 picks their favorite out of the two bundles in B'_t. Further, $|v_1(B'_{t,1}) - v_1(B'_{t,2})| \geqslant |v_1(B_{t,1}) - v_1(B_{t,2})|$ because $(B_{t,1}, B_{t,2})$ is the partition that minimizes the difference between agent 1's value for the two bundles. Putting the two together, we get the desired result. □

Next, we derive the same result for EF1+PO. Interestingly, we use the two cancelling EFX allocations produced for the previous result in order to show the existence of two cancelling EF1+PO allocations. This requires adding a minor tie-breaking rule to the procedure for computing these EFX allocations (the CUTANDCHOOSE++algorithm due to Plaut and Roughgarden [2018]): when the chooser (agent 2 in B_t) is indifferent between the two bundles but the cutter (agent 1 in B_t) is not, the chooser must pick the bundle the cutter values less. An interested reader can note that the allocations would have remained EFX even if we had introduced this tie-breaking in the proof of Theorem 10, but it was not needed there.

Definition 10 (Pareto Optimality (PO))**.** *An allocation A of a set of goods S is* Pareto optimal *(PO) if there is no allocation A' such that $v_i(A'_i) \geqslant v_i(A_i)$ for all $i \in N$ and at least one inequality is strict.*

Theorem 11. *For temporal fair division with $n = 2$ agents, an allocation that is EF1+PO per day and EF1 up to each day exists.*

Proof. Due to Lemma 4, it is sufficient to prove that for any day t, there exist EF1+PO allocations B_t and B'_t of the goods in M_t that cancel out.

We claim that for any day t, the existence of 2 EFX allocations that cancel out implies the existence of 2 EF1+PO allocations that cancel out. For contradiction, assume this were false, and for some instance there are not 2 EF1+PO allocations that cancel out.

Let the allocation B_t be the allocation that was constructed by the process in Theorem 10, where the bundles are chosen according to agent 1's valuations, and agent 2 picks their preferred bundle. We know that in this allocation, $v_2(B_{t,2}) \geqslant v_2(B_{t,1})$, and for all $g \in B_{t,2}, v_1(B_{t,1}) \geqslant v_1(B_{t,2} \setminus \{g\})$. If B_t is a PO allocation, then we do not have to deal with it further, otherwise, we know that there exists some $P \subseteq B_{t,1}, Q \subseteq B_{t,2}$, such that the allocation $((B_{t,1} \setminus P) \cup Q, (B_{t,2} \setminus Q) \cup P)$ is PO, and that $v_1((B_{t,1} \setminus P) \cup Q) \geqslant v_1(B_{t,1})$ and $v_2((B_{t,2} \setminus Q) \cup P) \geqslant v_2(B_{t,2})$.

First, note that in this PO allocation Q must be a strict subset of $B_{t,2}$. If this were not true, then $v_2((B_{t,2} \setminus Q) \cup P) \geqslant v_2(B_{t,2})$ would tell us that $v_2(B_{t,1}) \geqslant v_2(P) \geqslant v_2(B_{t,2})$, with the first inequality being due to the fact that $P \subseteq B_{t,1}$. By the process used to construct B_t, we already know that $v_2(B_{t,2}) \geqslant v_2(B_{t,1})$, meaning $v_2(B_{t,2}) = v_2(B_{t,1})$, and by the tie-breaking mechanism in the cut-and-choose algorithm, this would mean that $((B_{t,1} \setminus P) \cup Q, (B_{t,2} \setminus Q) \cup P)$ must be an EF+PO allocation. This would lead to a contradiction, since 2 copies of any EF allocation clearly cancel out.

Therefore, we must have that $Q \subset B_{t,2}$. Due to this, we know that there will be some good g such that $g \in (B_{t,2} \setminus Q) \cup P$ and $g \in B_{t,2}$. After the reallocation, we know that no agent was made worse off, so it will be the case that Agent 2 will not feel envy in the new PO allocation. We know that $((B_{t,1} \setminus P) \cup Q, (B_{t,2} \setminus Q) \cup P)$ cannot be an EF allocation (by the logic from the above paragraph this would lead to contradiction), so we know that Agent 1 does feel envy in the PO allocation. Since B_t is EFX, we know that for all $g' \in B_{t,2}$, we have that $v_1(B_{t,1}) \geqslant v_1(B_{t,2}) - v_1(g')$. Since we only have 2 agents, we have that $v_1(B_{t,2}) = v_1(M_t) - v_1(B_{t,1})$, and thus $v_1(g') \geqslant v_1(M_t) - 2v_1(B_{t,1})$ for all $g' \in B_{t,2}$. Therefore, due to the fact that $v_1((B_{t,1} \setminus P) \cup Q) \geqslant v_1(B_{t,1})$, we have that $v_1(g) \geqslant v_1(M') - 2v_1((B_{t,1} \setminus P) \cup Q)$, which can be rearranged back into $v_1((B_{t,1} \setminus P) \cup Q) \geqslant v_1((B_{t,2} \setminus Q) \cup P) - v_1(g)$, showing that this PO allocation is also EF1.

We can symmetrically repeat this procedure with the other EFX allocation B'_t where the bundles are selected according to Agent 2's valuations and Agent 1 chooses their preferred bundle. Since our PO reallocation can only increase the utility of both agents (and thus lessen their envy), we can conclude that if the original EFX allocations cancel out, the corresponding EF1+PO allocations also cancel out, giving a contradiction. □

We note that while we provided a polynomial-time algorithm in Theorem 3 for achieving SD-EF1 per day and EF1 up to each day result, the constructions in Theorems 10 and 11 for achieving EFX or EF1+PO per day are not efficient because they rely on partitioning a set of numbers into two subsets with near-equal sum. This is NP-hard because PARTITION (which requires exactly equal sum) can be trivially reduced to it. This raises the following interesting open question:

Open Question 2: For temporal fair division with $n = 2$ agents, can EFX or EF1+PO per day and EF1 up to each day be achieved in polynomial time?

D.5 Proof of Theorem 4

Proof of Theorem 4. Consider the following instance in which four goods arrive over three days: $M_1 = \{g_1, g_4\}$, $M_2 = \{g_3\}$, and $M_3 = \{g_2\}$. Two agents have identical valuations given by $v(g_1) = 4$, $v(g_2) = 3$, $v(g_3) = 2$, and $v(g_4) = 1$ (since we seek SD-EF1, only the fact that the agents strictly prefer $g_1 \succ g_2 \succ g_3 \succ g_4$ matters).

For an allocation A to be SD-EF1 up to each day, $A_{\overline{M}_t}$ must be SD-EF1 for each $t \in [3]$, where $\overline{M}_1 = \{g_1, g_4\}, \overline{M}_2 = \{g_1, g_3, g_4\}$, and $\overline{M}_3 = M = \{g_1, g_2, g_3, g_4\}$.

By the necessity condition of Proposition 1, we can make the following claims:

- For $A_{\overline{M}_1}$ to be SD-EF1, g_1 and g_4 must be given to different agents.
- For $A_{\overline{M}_2}$ to be SD-EF1, g_1 and g_3 must be given to different agents.
- For $A_{\overline{M}_3}$ to be SD-EF1, the two goods in $T(M, 2) = \{g_1, g_2\}$ must be allocated to different agents. Since each agent must also get 2 goods from $T(M, 4) = M$, it follows that g_3 and g_4 must also be allocated to different agents.

We now have the requirements that g_1, g_3, and g_4 must all be pairwise given to different agents, which is impossible since there are only two agents. Hence, in this instance, there is no allocation that is SD-EF1 up to each day. □

D.6 Proof of Theorem 5

Proof. Consider the following instance, in which eight goods arrive over three days: $M_1 = \{g_1, g_5, g_7\}$, $M_2 = \{g_2, g_4, g_6\}$, and $M_3 = \{g_3, g_8\}$. The valuations are as follows:

$$\begin{array}{cccccccc} v_1(g_1) = 8 & v_1(g_2) = 7 & v_1(g_3) = 6 & v_1(g_4) = 5 & v_1(g_5) = 4 & v_1(g_6) = 3 & v_1(g_7) = 2 & v_1(g_8) = 1 \\ v_2(g_2) = 8 & v_2(g_3) = 7 & v_2(g_1) = 6 & v_2(g_4) = 5 & v_2(g_6) = 4 & v_2(g_8) = 3 & v_2(g_5) = 2 & v_2(g_7) = 1 \end{array}$$

Notice that although each agent has a different strict ordering over M, their orderings restricted to the goods in any day M_t are identical. Since there are only 2 or 3 goods given on any day and each agent has strict preferences over them, for allocation A_{M_t} to be EF1, the same agent cannot receive both the items from $T(M_t, 2)$ (otherwise, the other agent, who receives at most one good from M_t that they value strictly less than each good in $T(M_t, 2)$, would envy them even after the removal of one of the goods).

Thus, EF1 per day requires that among the pairs of goods (g_1, g_5), (g_2, g_4), and (g_3, g_8), the two goods in each pair go to different agents.

Now consider what is required in order for the allocation A_M to be SD-EF1 overall while adhering to these constraints. Since both agents have strict preferences over the goods, by the necessity condition of Proposition 1, we can say that for both $i \in \{1, 2\}$, $|A_{M,i} \cap T_i(M, r)| \geqslant \lfloor r/n \rfloor$ must be true for all $r \in [|M|]$.

This means that each agent must receive at least one of their top 2 goods. Specifically, agent 1 must receive one of $T_1(M, 2) = \{g_1, g_2\}$, and agent 2 must receive at least one of $T_2(M, 2) = \{g_2, g_3\}$.

It also means that each agent must receive at least 2 of their top 4 goods. Observe that $T_i(M, 4) = \{g_1, g_2, g_3, g_4\}$ for both agents, so, in order to satisfy this, each agent must receive *exactly* 2 goods from $\{g_1, g_2, g_3, g_4\}$.

Notice that under these restrictions, in any allocation where agent 1 receives g_2, they cannot receive g_3 (or they would have both of agent 2's top 2 goods) or g_4 (due to the per day constraints). Since the allocation over the top 4 goods must be balanced (each agent receiving exactly two of them), the only possible allocation $A_{T_1(M,4)}$ in this scenario would be $(\{g_1, g_2\}, \{g_3, g_4\})$. Using the same logic, when agent 2 receives g_2, the only possible allocation is $(\{g_1, g_4\}, \{g_2, g_3\})$. Since one of the agents must be given g_2, these are the only two ways that the top 4 goods can be allocated. Note that in both of these allocations, agent 1 gets g_1 and agent 2 gets g_3.

Finally, consider how the remaining goods $\{g_5, g_6, g_7, g_8\}$ must be allocated to guarantee SD-EF1 overall. Each agent must receive at least 3 of their top 6 goods, and since we know that each agent has exactly 2 of their top 4 goods, that means that agent 1 must receive at least one of $\{g_5, g_6\}$, and agent 2 must receive at least one of $\{g_6, g_8\}$. Since we know that agent 1 must be allocated g_1, the per-day constraints say they cannot receive g_5, so they must receive g_6. Similarly, agent 2 is known to have g_3, so they cannot receive g_8, which means they must also receive g_6, which is a contradiction. Hence, in this instance, there is no allocation that is EF1 per day and SD-EF1 overall. □

D.7 Two Agents and Identical Days

In this section, we briefly discuss the implications of our results so far for the special case where we have two agents and identical days. Recall that Igarashi *et al.* [2024] focus only on the case of identical days and many of their results hold for only two agents. In particular, they show that an allocation that is EF1 per day and (exact) EF overall exists and can be computed in polynomial time when the number of days k is even.

We prove a slightly stronger result via a much simpler technique, albeit only for allocating goods while their result is for allocating a mixture of goods and chores.

Theorem 12. *For temporal fair division with $n = 2$ agents and identical days, an allocation that is SD-EF1 per day, SD-EF1 up to each day, and SD-EF up to each even day exists and can be computed in polynomial time.*

Proof of Theorem 6. Consider the set of goods M_1 on day 1 (each day has a set of goods identical to this). Consider the two allocations $B = (B_1, B_2)$ and $B' = (B_2, B_1)$ produced in Lemma 5 such that both B and B' are SD-EF1 allocations of M_1. The desired allocation is one that uses B on every odd day and B' on every even day. Clearly, SD-EF1 per day is satisfied. Since the allocations completely cancel out after every even day (each agent has exactly the same number of copies of each good), we get SD-EF1 up to each day and SD-EF up to each even day. □

E Missing Proofs from Section 5

E.1 Proof of Theorem 7

Proof of Theorem 7. Start by constructing two set families, P_1 and P_2.

$$P_1 = \{T(M_t, nr) \setminus T(M_t, n(r-1)) : r \in \left\lceil \frac{|M_t|}{n} \right\rceil, t \in [k]\},$$
$$P_2 = \{T(M, nr) \setminus T(M, n(r-1)) : r \in \left\lceil \frac{|M|}{n} \right\rceil\}.$$

In words, P_2 splits the entire set of goods M into the agents' most preferred n goods, their next most preferred n goods, etc. P_1 does a similar partitioning, but splits the goods from each day separately.

Because preferences orderings of all agents are identical, we can use the sufficiency condition Proposition 1, which states that if for all $i \in N$ and $r \in [|M|]$, $|A_i \cap T_i(M, r)| \in \{\lfloor r/n \rfloor, \lceil r/n \rceil\}$ implies SD-EF1 overall. Clearly, if each agent receives at most 1 good from each set in P_2, this will be true. Similarly, if each agent receives at most 1 good from each set in P_1,

then for all $i \in N$, $t \in [k]$, and $r \in [|M_t|]$, we would have $|A_i \cap T_i(M_t, r)| \in \{\lfloor r/n \rfloor, \lceil r/n \rceil\}$, implying SD-EF1 per day. Therefore, any allocation which gives each agent at most 1 good from each of the sets in $P_1 \cup P_2$ will be SD-EF1 per day and SD-EF1 overall. Our goal is to find an allocation that meets these constraints.

We will first start by adjusting the structure of the set families slightly to make the problem easier to work with. Note that it must be the case that $|P_1| \geqslant |P_2|$. P_2 will contain $\lceil m/n \rceil$ sets. $\lfloor m/n \rfloor$ set of exactly n goods, and if m/n is not an integer, then 1 additional set will be included containing the remaining goods. P_1 will contain at least $\lceil m/n \rceil$ sets, since it will be a disjoint partition of all the goods in m, with each set having a maximum set size of n, but may contain up to k sets with size less than n due to the fact that it is partitioning each day individually.

First, we will create $|P_1| - |P_2|$ empty sets and place them in P_2. This makes sure that P_1 and P_2 have the same number of sets. Then, if $|P_1| > \frac{m}{n}$, we will create $n|P_1| - m$ dummy goods (goods with 0 value to every agent), place each dummy good into a set from P_1 and P_2 in such a way that all $|P_1|$ sets in both families contain exactly n goods each. Note that the restriction that each agent must receive no more than 1 good from each of these updated families creates is a stronger constraint than what is needed to guarantee SD-EF1. Any allocations that meets the constraints on the updated set families can be clearly shown to meet the original SD-EF1 constraints simply by removing all the dummy goods. After these updates, both P_1 and P_2 will contain the same number of sets, and the size of every set in both families will be exactly n.

Next, construct a graph where each set from P_1 or P_2 is a vertex, and put an edge between two sets in P_1 and P_2 if they share a good (there can be multiple edges between vertices). Label each edge to correspond to the good it represents. Since both P_1 and P_2 are disjoint partitions over the set of goods M + "the dummy goods", there will be no edge between two sets from the same family, making this a bipartite graph where each vertex has a degree of exactly n. Therefore, this bipartite graph can be deconstructed into n perfect matching. Let each of these matchings denote a bundle of the items corresponding to the edges in that matching. These n bundles will form a disjoint partition over the full set of goods M, and each bundle will contain at most 1 good from each set in $P_1 \cup P_2$.

Since there are n bundles, we can create an allocation by assigning each one arbitrarily to an agent. Any allocation formed this way will meet the constraints for both P_1 and P_2, and thus will be be SD-EF1 per day and SD-EF1 overall. □

E.2 Proof of Theorem 8

Proof of Theorem 8. Consider the following instance in which twenty-four goods arrive over four identical days: $M_t = \{g_{1,t}, g_{2,t}, g_{3,t}, g_{4,t}, g_{5,t}, g_{6,t}\}$ for all $t \in [4]$. Twelve agents have identical valuations given by $v(g_{1,t}) = 6$, $v(g_{2,t}) = 5$, $v(g_{3,t}) = 4$, $v(g_{4,t}) = 3$, $v(g_{5,t}) = 2$, $v(g_{6,t}) = 1$ (since we seek SD-EF1, it only matters that the agents strictly prefer $g_{1,t} \succ g_{2,t} \succ g_{3,t} \succ g_{4,t} \succ g_{5,t} \succ g_{6,t}$ for all $t \in [4]$, and are indifferent between any two goods $g_{l,t}, g_{l,t'}$ for all $l \in [6], t, t' \in [4]$.

Consider how the requirement of SD-EF1 up to each day restricts how each good can be allocated:

- The bundle $\overline{M}_2$ will contain 12 goods. By the necessity condition of Proposition 1, we know for $A_{\overline{M}_2}$ to satisfy SD-EF1, it must be the case that each Agent receives exactly 1 good from $T(\overline{M}_2, 12) = \overline{M}_2$.
- For $A_{\overline{M}_3}$ to be SD-EF1, it must be the case that each agent receives exactly 1 of the goods in $T(\overline{M}_3, 12)$. This is because $T(\overline{M}_3, 12)$ will contain goods $g_{1,t}, g_{2,t}, g_{3,t}, g_{4,t}$ for all $t \in [3]$. The next good in the agents' ordering will be a copy of $g_{5,t}$ for some t, so all the goods in $T(\overline{M}_3, 12)$ are strictly preferred over the remaining goods in $\overline{M}_3 \setminus T(\overline{M}_3, 12)$.
- For $A_{\overline{M}_4}$ to be SD-EF1, it must be the case that each agent gets exactly 1 of the goods in $T(\overline{M}_4, 12)$. This is because $T(\overline{M}_4, 12)$ will contain goods $g_{1,t}, g_{2,t}, g_{3,t}$ for all $t \in [4]$. Similar to the previous case, the next good in the agents' ordering will be a copy of $g_{4,t}$ for some t, so it must be the case that all the goods in $T(\overline{M}_4, 12)$ are strictly preferred over the remaining goods in $\overline{M}_4 \setminus T(\overline{M}_4, 12)$. Since there are 24 total goods in $\overline{M}_4$, we can also say that each agent must receive exactly 2 goods from the set $T(\overline{M}_4, 24) = \overline{M}_4$. Since no agent can have more than 1 good from $T(\overline{M}_4, 12)$, the only way to satisfy this condition is to give each agent exactly 1 good from $T(\overline{M}_4, 24) \setminus T(\overline{M}_4, 12)$.

Figure 2 makes these restrictions clear. The restrictions over $\overline{M}_2$ are shown in Blue, $\overline{M}_3$ in Green, and $\overline{M}_4$ in Red. Each box represents a group of 12 goods that all must go different agents. Notice that the good $g_{4,3}$ is contained in a restriction from $\overline{M}_3$ and $\overline{M}_4$. Due to its $\overline{M}_3$ restriction, it cannot be given to an agent that has been assigned a good from $T(\overline{M}_2, 8)$. Due to its $\overline{M}_4$ restriction, it cannot be given to an agent that been assigned a good from $T(\overline{M}, 12) \setminus T(\overline{M}_2, 6)$. However, these two groups together make up the entire set of goods $\overline{M}_2$, which contains 12 goods, each assigned to one of the 12 agents. So there is no agent we can assign $g_{4,3}$ to that will lead to the satisfaction of SD-EF1 for $A_{\overline{M}_2}$, $A_{\overline{M}_3}$ and $A_{\overline{M}_4}$. □

F Missing Proofs from Section 6

F.1 Helper Lemmas for Proving Theorem 9

Lemma 6. *For any instance with identical days, the identical orderings transformation described in Algorithm 1 will produce an instance where all agents have identical orderings overall.*

	$t=1$	$t=2$	$t=3$	$t=4$
g_1	1	7	/	/
g_2	2	8	/	/
g_3	3	9	/	/
g_4	4	10	/	/
g_5	5	11	/	/
g_6	6	12	/	/

Figure 2: Restrictions Placed on the Allocation by SD-EF1 Up To Any Day

Proof. Let M' represent the full set of goods in the identical orderings transformation instance, and for each agent i, let v'_i be i's valuations over M'. Due to the original instance having identical days, we know that for each pair of days t, t' there must be a bijection $f : M_t \to M_{t'}$ such that for each $g \in M_t$ and each $i \in N$, $v_i(g) = v_i(f(g))$. By the way we constructed the identical orderings transformation instance, we know that there must be an analogous bijection in the transformed instance $f' : M'_t \to M'_{t'}$. Let $M'_t = \{g_{t,1}, \ldots, g_{t,|M'_t|}\}$, where for each $r, r' \in [|M_t|]$ with $r' \geqslant r$, we have that $g_{t,r} \succcurlyeq g_{t,r'}$ for all agents. Without loss of generality, we can assume that the labeling on each day is consistent, and for each day t, t', the bijection f' maps $g_{t,r}$ to $g_{t',r}$ for all $r \in [|M_t|]$. Therefore, can assume that the overall ordering of $g_{1,1} \succcurlyeq_i g1,2 \succcurlyeq_i \ldots \succcurlyeq_i g_{1,k} \succcurlyeq_i g_{2,1} \succcurlyeq_i \ldots \succcurlyeq_i g_{|M_k|,k}$ must hold. □

Lemma 7. *The algorithm described in Section 6 achieves SD-PROP1 Overall.*

Proof. This proof involves reasoning with two separate bijections. The first is $o_i : M \to M'$, which is the bijection formed by creating the identical ordering transformation in Algorithm 1. In this bijection, we have that $v_i(g) = v'_i(o_i(g))$ for each $i \in N$ and $g \in M$, which implies that $v_i(M) = v'_i(M')$ for all i, and that the utility vectors formed by taking all of i's utilities for M and M', and ordering them non-increasingly, will both be identical. The second is $z : A_i \cup \{g_i\} \to A'_i \cup \{g'_i\}$, where A_i and A'_i are the bundles that i receives in the final allocation and the identical orderings transformation allocation of Algorithm 1 respectively, and g_i, g'_i are agent i's favourite good that they are not allocated in each of those allocations. In this bijection, we will have that $v_i(g) \geqslant v'_i(z(g))$ for all $i \in N$ and $g \in A_i \cup \{g_i\}$, implying that $v_i(A_i \cup \{g_i\}) \geqslant v'_i(A'_i \cup \{g'_i\})$.

Since SD-EF1 implies EF1, and the allocation found over the transformed preferences reduction by this algorithm will be SD-EF1 overall, it is easy to see that we can invoke the same logic from Lemma 2 and Lemma 3 to show that this allocation is PROP1 overall. To show that the allocation is in fact SD-PROP1 overall, we can do a more detailed analysis on the proof technique from Lemma 3. Consider the set of goods $M^* = M \cup M'$ and each agent i's ordering over M^* that is induced by their valuation functions v_i and v'_i. We note that when considering the set M^*, the bijection from Lemma 3 not only shows that $v_i(A_i \cup \{g_i\}) \geqslant v'_i(A'_i \cup \{g'_i\})$, but also allows us to conclude the stronger claim that $A_i \cup \{g_i\} \succcurlyeq_i^{\text{SD}} A'_i \cup \{g'_i\}$. This is because for each pair of items $g^*, z(g^*)$, if we have that $v_i(g^*) \geqslant v'_i(z(g^*))$, then we must also have $g^* \succcurlyeq_i z(g^*)$.

Next, note that for each agent $i \in N$, and each $g \in M$, we must have that $H_i(M', o_i(g)) = \{o_i(g') : g' \in H_i(M, g)\}$, since clearly, $H_i(M', o(g))$ can be thought of as the set of all goods $g^* \in M'$ such that $g^* \succcurlyeq_i o_i(g)$, and similarly, $H_i(M, g)$ is the set of goods $g^* \in M$ such that $g^* \succcurlyeq_i g$. From our definition of o_i, we have that for any $g^* \in M$, $v_i(g^*) \geqslant v_i(g)$ if and only if $v'_i(o_i(g^*)) \geqslant v'(o_i(g))$.

With this, we can conclude that $|H_i(M, g)| = |H_i(M', o_i(g))|$, and also that $|(A_i \cup \{g_i\}) \cap H_i(M, g)| \geqslant |(A'_i \cup \{g'_i\}) \cap H_i(M', o_i(g))|$ for all goods $g \in M$. The second statement follows from the fact that for any $g^* \in A_i \cup \{g_i\}$, if $z(g^*) \in H_i(M', o_i(g))$, then $g^* \in H_i(M, g)$ must be true as well, since we have that $v_i(g^*) \geqslant v'_i(z(g^*))$, and $v_i(g) = v'_i(o_i(g))$.

We know that A'_i is SD-EF1, implying it is SD-PROP1 by Proposition 2. Therefore, for all $g \in M$, we have that $|(A_i \cup \{g_i\}) \cap H_i(M, g)| \geqslant |(A'_i \cup \{g'_i\} \cap H_i(M', o_i(g))| \geqslant \lceil |H_i(M', o_i(g))|/n \rceil = \lceil |H_i(M, g)|/n \rceil$, allowing us to conclude that A is SD-PROP1. □

F.2 Proof of Theorem 9

Proof of Theorem 9. From Lemma 6, it is known that the identical days reduction will construct an instance where each agent has the same overall ordering for the full set of goods M'. This means that by the results of Theorem 7, an allocation A' over M' can be found that is SD-EF1 per day and SD-EF1 overall in polynomial time.

Using identical logic from Lemma 1, constructing the final allocation A by running the picking order construction procedure from Algorithm 1 on A' will result in an allocation that is SD-EF1 per day, since A' is SD-EF1 per day.

Finally, from Lemma 7, we know that A will also be SD-PROP1 overall, completing the proof. □

G Laminar Fair Division

For any set of goods M, a collection $\mathcal{L}$ of subsets of M is a Laminar Set Family over M if for every pair of subsets $S, T \in \mathcal{L}$, we have that either $S \cap T = \emptyset$, $S \subset T$, or $T \subset S$.

For each set $S \in \mathcal{L}$, let $D(S) : \mathcal{L} \to 2^{\mathcal{L}}$ be a function returning each set $S' \in \mathcal{L}$ such that $S' \subset S$. Let $C(S) : \mathcal{L} \to 2^{\mathcal{L}}$ be the function returning every maximal set of $D(S)$ (every set in $D(S)$ that is not contained in some other set from $D(S)$).

We can say that a laminar set family $\mathcal{L}$ over M is *Complete* if and only if the following conditions hold:

- The set M is an element of $\mathcal{L}$.
- For every set $S \in \mathcal{L}$, either $D(S) = \emptyset$, or $\cup_{S' \in D(S)} = S$.

We will assume that all laminar sets families we deal with in our setting are complete. This can be assumed without loss of generality, as any laminar set $\mathcal{L}$ over a set of goods M that is not complete can be "completed" through the following simple procedure:

- If the full set of goods M is not in $\mathcal{L}$, add M to $\mathcal{L}$.
- For every set $S \in \mathcal{L}$, if $0 < | \cup_{S' \in D(S)} | < |S|$, then create a new set $S^* = S \setminus \cup_{S' \in D(S)}$ and add it to $\mathcal{L}$.

After the above steps, $\mathcal{L}$ will remain a laminar set family. $\mathcal{L}$ will clearly remain laminar after the addition of M since every other set in $\mathcal{L}$ will be a strict subset of M. $\mathcal{L}$ will also remain laminar after the addition of each of the S^* sets, since the definition of laminar families ensures that if none of the sets in $\mathcal{L}$ that are subsets of a set S contain some good g, then the only other sets in the family that can contain g are strict supersets of S. Therefore, the S^* that the completion process adds to $\mathcal{L}$ will be a strict subset of S, and thus a strict subset of all the sets containing g.

Note that in any complete laminar set family $\mathcal{L}$, and any $S \in \mathcal{L}$, $C(S)$ will either be empty, or will form a complete disjoint partitioning of S. If this were not true, then we would have that $C(S) \neq \emptyset$, which implies that $D(S) \neq \emptyset$. We would also have that $\cup_{S' \in C(S)} \subset S$ and $\cup_{S' \in D(S)} = S$. Clearly, every good $g \in \cup_{S' \in D(S)}$ must appear in some maximal set from $D(S)$, giving a contradiction. The fact that all the sets in $C(S)$ will be disjoint follows immediately from the definition of a laminar set family and from the fact that each set $S' \in C(S)$ is maximal in $D(S)$.

One can think of the structure of a complete laminar set family $\mathcal{L}$ as a tree where each subset is a node. M is the root node of the tree. For any set $S \in \mathcal{L}$, $D(S)$ are the **descendants** on S, and $C(S)$ are the **children** of S. The **Leaf sets** of $\mathcal{L}$ are any sets $S \in \mathcal{L}$ who has no descendants (i.e. $D(S) = \emptyset$). Thinking of complete Laminar Set Families in this way will allow us to topographically sort the sets. Particularly, thinking of $\mathcal{L}$ as a directed tree where there is a directed edge going from each child to its parent, then a topographical sorting of $\mathcal{L}$ will result in no set $S \in \mathcal{L}$ appearing in the ordering before any of its descendants.

Now, consider the following problem. A school district has received a new shipment of supplies and must distribute them between the schools under their jurisdiction. Each item is used for a different subject (there are microscopes for the biology lab, instruments for music class, easels and paint for art class, etc.), and each school has different preferences over the items. The district wants to find an allocation that is fair among the schools, but also wants to make sure that the allocations are fair with respect to each individual subject. For example, the district may be able to construct an EF1 allocation by giving all the biology supplies to one school, and all the art supplies to another school, but that would be extremely unfair to the individual departments within those schools. Interestingly, this problem is identical to finding an allocation over a temporal instance that is fair per day and fair overall, since both are in essence looking at a pairwise-disjoint partition of some set of items M, and finding an allocation that is fair with respect to M, and remains fair when only looking at any of the sets in the partition.

Up until this point, the best way to solve such a problem would be existing algorithms for constrained fair division, such as the cardinality constraint algorithm of [Biswas and Barman, 2018]. This algorithm would guarantee an EF1 overall allocation, while also guaranteeing that each school got a balanced number of items from each subject, but it gives no guarantee that the goods from each subject will be allocated fairly according to the valuation functions of each of the schools. While our solution, Algorithm 1, only achieves PROP1 overall, it achieves the very strong SD-EF1 fairness guarantee for each subject, making it arguably a more desirable algorithm for use cases such as this. When there are 2 agents, or when all agents' orderings are identical, then we get strictly better guarantees, since we can still achieve SD-EF1 per day (which implies balancedness), while also achieving EF1 overall.

While the "per day" and "overall" desiderata translate very well into this broader interpretation of the temporal fairness model, our other definition, "up to each day" does not. This is because "up to each day" implicitly assumes that there is some ordering over the sets of goods in the partition. One intuitive way to extend this concept is with laminar set families. In the school district example, the sets of items where fairness is required could be something like $\{\text{Biology}, \text{Chemistry}, \text{Drawing}, \text{Music}, \text{Science}, \text{Arts}\}$, where Biology and Chemistry are subsets of Science and Drawing and Music are subsets of Arts.

With this in mind, we can introduce a new concept of fairness, which generalizes "up to each day" and is more compatible with this abstract view of temporal fair division.

Definition 11 (Laminar Fairness)**.** *For desideratum X, allocation A is Laminar X with respect to some laminar set family $\mathcal{L}$ if A_S satisfies X for all $S \in \mathcal{L}$.*

It can be seen that achieving both "up to each day" and "per day" fairness is a special case of Laminar Fairness. Figure 3 shows the laminar family induced by these constraints.

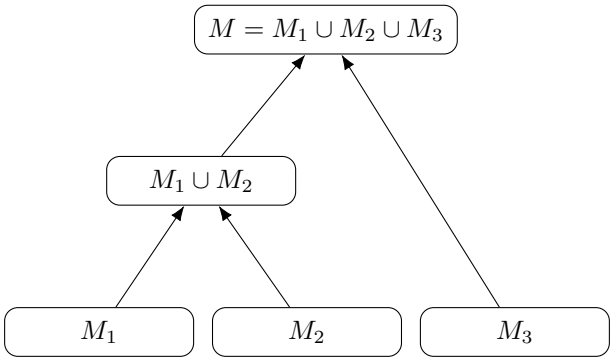

Figure 3: Representation of the "up to each day" and "per day" constraints for three days as a laminar set family. An allocation that is EF1 up to each day and EF1 per day would be an allocation that is EF1 with respect to every set in this family. To just represent the "up to each day" constraints, only the sets M_1, $M_1 \cup M_2$ and $M_1 \cup M_2 \cup M_3$ would be required.

We present an upgrade to our two agent Envy-Balancing algorithm from Lemma 4 that allows us to find EF1 allocations with respect to any laminar set family, a strictly stronger guarantee than simply finding allocations that are EF1 up to each day.

Theorem 13. *Given 2 agents, a set of goods M, and a laminar set family $\mathcal{L}$ over M, it is possible to find an allocation to those agents that is Laminar EF1 with respect to $\mathcal{L}$.*

Proof. We begin by introducing a slight adaptation of the Envy-Balancing algorithm from Lemma 4, which will serve as the basis for our algorithm to give the stronger guarantee of laminar fairness. We will use the notation introduced in the proof of Lemma 4 to descibe the behaviour of the algorithm.

At the end of the Algorithm 2, the allocation A induced by $F \cup S$ is returned as the final allocation. A is an EF1 allocation since we know that A_F is an EF allocation and A_S is EF1. Note that $A_{F \cup \text{SWAP}(S)}$ would also be an EF1 allocation for the same reason (The proof of Lemma 4 shows that at every point in Algorithm 2, both A_S and $A_{\text{SWAP}(S)}$ are EF1 allocations). Also note that the two allocations $A_{F \cup S}$ and $A_{F \cup \text{SWAP}(S)}$ will cancel out. This is because we know that neither agent feel envy in A_F, and we also know that A_S, and $A_{\text{SWAP}(S)}$ will cancel out for both $i \in \{1, 2\}$, due to the fact that a SWAP will involve replacing every allocation in S with an allocation that cancels it out, so the sum of all these allocations must also cancel out.

We will use this fact in order to construct Algorithm 3, named EnvyBalancing++. This algorithm takes as input a set of goods S, and a partitioning C over the goods in S. If the partitioning is empty, then EnvyBalancing++ simply returns 2 EF1 allocations over S that cancel out, using any of the methods introduced in Section 4 to do so. Otherwise, if C is a complete and disjoint partition of S, then EnvyBalancing++ runs Algorithm 2 on the input, but returns both $A_{F \cup S}$ and $A_{F \cup \text{SWAP}(S)}$ as the final allocations.

Algorithm 4 takes in a laminar set family $\mathcal{L}$, and finds an allocation that is Laminar EF1 with respect to $\mathcal{L}$ by the following process:

- Sort the sets from $\mathcal{L}$ topographically such that no set appears in the order before any of its descendants.
- Run EnvyBalancing++ on each set $S \in \mathcal{L}$ in the topographical order, with the partitioning of S being giving by the pairs of allocations the algorithm has already found for each of the children of S. If S is a leaf set, then $C(S) = \emptyset$, and the algorithm will return two arbitrary EF1 allocations over S that cancel out. Otherwise, two allocations generated by the Envy-Balancing algorithm will be returned.
- As the final output, return the 2 allocations over M, which will be the last set from $\mathcal{L}$ ordered topographically. Each of these allocations are guaranteed to be EF1 with respect to every set $S \in \mathcal{L}$.

We will prove the following inductive statement: For any set $S \in \mathcal{L}$, if the algorithm found 2 allocations for each child $S' \in C(S)$ that are EF1 with respect to S' and all the descendants of S', then the algorithm will find 2 allocations over S that are EF1 with respect to S and all descendants of S.

We will first start with the case where S is not a Leaf set. Since the algorithm visits each descendant of S prior to visiting S, it will have already found 2 allocations for every set $S' \in C(S)$. If the hypothesis holds, then each of these pairs of allocations will cancel out, will be EF1 with respect to S', and will be EF1 with respect to every descendent of S'. The algorithm will use these allocations as input to EnvyBalancing++. Since we are assuming that $\mathcal{L}$ is a complete laminar set family, the children of S will make a complete and disjoint partition of S. We know that EnvyBalancing++ will output two allocations over S that cancel out and are EF1 over S. Further, we know that both allocations will be EF1 with respect to all descendants of S due

to the fact that the fact that the Envy-Balancing algorithm constructs its allocations by picking one of the inputted allocations from each of its children, which are known to be EF1 with respect to all their descendants from the hypothesis.

If S is a Leaf set, the algorithm uses any arbitrary method to find the two allocations for S. These allocations will be EF1 with respect to S, and since leaf sets do not have any descendants, they will vacuously be EF1 with respect to all descendants as well. This proves the base case and completes the argument.

Finally, note that the topographical sorting of the sets runs in polynomial-time in the number of elements in the tree it is traversing, and it is well-known that every laminar set family with a ground set M can have at most $2|M| - 1$ members. This together with the fact that EnvyBalancing++ runs in poly(m) time allows us to conclude that the entire procedure will be poly(m). □

We give the proof of this lemma in Appendix G.

Unfortunately, for the general case, laminar fairness will not always be possible. Since the "up to each day" constraint can be modeled as a laminar set family, all of the impossibility results for "up to each day" also hold for laminar fairness.

Algorithm 3 EnvyBalancing++

Input for $t \in [k]$, a pair of allocations (B_t^1, B_t^2) of the set of goods M_t that cancel out, with labels assigned in such a way that if neither of the allocations is Envy-Free, then $v_1(B_{t,1}^1) - v_1(B_{t,2}^1) \geqslant 0$ and $v_2(B_{t,2}^2) - v_2(B_{t,1}^2) \geqslant 0$.

Output An allocation A of the set of goods $M = \cup_{t \in [k]} M_t$

```
if C = ∅ then
    (A, A') ← Two EF1 allocations of S that cancel out, determined by some arbitrary subroutine
else
    F ← ∅, S ← ∅, e_1 ← 0, e_2 ← 0
    for t ∈ [k] do
        if v_1(B^1_{t,1}) ⩾ v_1(B^1_{t,2}) ∧ v_2(B^1_{t,2}) ⩾ v_2(B^1_{t,1}) then
            F ← F ∪ {B_t}
        else if v_1(B^2_{t,1}) ⩾ v_1(B^2_{t,2}) ∧ v_2(B^2_{t,2}) ⩾ v_2(B^2_{t,1}) then
            F ← F ∪ {B'_t}
        else
            if e_1 ⩽ 0 then
                S ← S ∪ {B^1_t}
                e_1 ← e_1 + (v_1(B^1_{t,1}) − v_1(B^1_{t,2}))
                e_2 ← e_2 + (v_2(B^1_{t,2}) − v_2(B^1_{t,1}))
            else
                S ← S ∪ {B^2_t}
                e_1 ← e_1 + (v_1(B^2_{t,1}) − v_1(B^2_{t,2}))
                e_2 ← e_2 + (v_2(B^2_{t,2}) − v_2(B^2_{t,1}))
            end if
        end if
        if e_1 ⩾ 0 ∧ e_2 ⩾ 0 then
            F ← F ∪ S
            S ← ∅, e_1 ← 0, e_2 ← 0
        else if e_1 ⩽ 0 ∧ e_2 ⩽ 0 then
            F ← F ∪ SWAP(S)
            S ← ∅, e_1 ← 0, e_2 ← 0
        end if
    end for
    A ← allocation in which M_t is allocated according to the allocation of M_t in F ∪ S, for each t ∈ [k]
    A' ← allocation in which M_t is allocated according to the allocation of M_t in F ∪ SWAP(S), for each t ∈ [k]
end if
return (A, A')
```

Algorithm 4 Laminar Envy-Balancing Algorithm

Input A Laminar Set $\mathcal{L}$ **Output** An allocation A of the set of goods $M = \cup_{S \in \mathcal{L}}$

$T \leftarrow (S_1, S_2, \ldots, S_{|\mathcal{L}|})$ (A topographical ordering the sets in $\mathcal{L}$)

$\forall S \in \mathcal{L}, B_S \leftarrow \emptyset$

for $t \in [|\mathcal{L}|]$ **do**

  $(B_S^1, B_S^2) \leftarrow$ EnvyBalancing++(S_t, $\{B_{S'} : S' \in C(S_t)\}$)

  $B_{S_t} = (B_S^1, B_S^2)$

end for

return $B_{S_{|\mathcal{L}|}}$