Fair Division with Indivisible Goods, Chores, and Cake

Haris Aziz[1], Xinhang Lu[2], Simon Mackenzie[1], and Mashbat Suzuki[1]

[1]UNSW Sydney, Australia, haris.aziz@unsw.edu.au,
simon.william.mackenzie@gmail.com, mashbat.suzuki@unsw.edu.au
[2]Kyushu University, Japan, xinhang.lu@inf.kyushu-u.ac.jp

Abstract

We study the problem of fairly allocating indivisible items and a desirable heterogeneous divisible good (i.e., cake) to agents with additive utilities. In our paper, each indivisible item can be a good that yields non-negative utilities to some agents and a chore that yields negative utilities to the other agents. Given a fixed set of divisible and indivisible resources, we investigate almost envy-free allocations, captured by the natural fairness concept of *envy-freeness for mixed resources (EFM)*. It requires that an agent i not envy another agent j if agent j's bundle contains any piece of cake yielding positive utility to agent i (i.e., envy-freeness), and that agent i be envy-free up to one item (EF1) towards agent j otherwise. We prove that with indivisible items and a cake, an EFM allocation always exists for any number of agents with additive utilities.

We reduce the problem of showing the existence of EFM to the problem of proving the compatibility of EF1 and envy-freeability. An allocation of indivisible items is said to be *envy-freeable* if the allocation can be made envy-free by allocating a sufficient amount of money to each of the agents. We show that for indivisible items, an EF1 and envy-freeable allocation always exists. Moreover, such an allocation can be found in polynomial time. This result, in and of itself, is of independent interest.

1 Introduction

Fair division studies the fundamental problem of allocating valuable resources or undesirable burdens among agents with possibly differing preferences in a fair manner. The problem was first formulated mathematically in the seminal work of Steinhaus [1949], which addresses the fair division of a heterogeneous divisible resource, commonly known as *cake cutting*. Since then, fair division has attracted ongoing interest in mathematics, political science, and economics [Brams and Taylor, 1996], and most recently has received extensive attention in computer science.

A thorough understanding of fairness has been described as a key to "addressing some of the greatest challenges of the 21st century" [Procaccia, 2013]. Among the plethora of fairness criteria proposed in the literature, *envy-freeness* has been regarded as the "gold standard" of fairness in allocation problems [e.g., Benadè et al., 2024]. It requires that each agent weakly prefer their own bundle to any other agent's bundle. An envy-free cake division always exists [Alon, 1987; Su, 1999], and can be found via a discrete and bounded protocol [Aziz and Mackenzie, 2016a,b] under the query model of Robertson and Webb [1998].

Looking beyond cake cutting, the allocation of heterogeneous *indivisible* items is of practical importance and has attracted considerable attention over the past decades, especially in the context of allocating goods (items with non-negative utilities) [Amanatidis et al., 2023; Nguyen

and Rothe, 2023; Suksompong, 2021, 2025]. More recently, due to its broad applicability, the fair allocation of chores (items that yield negative utilities) has also received significant attention [Guo et al., 2023].

There has been a concerted effort to understand whether strong and compelling fairness guarantees can be achieved in the more general setting that includes a mix of both goods and chores [e.g., Aziz et al., 2020, 2022, 2027; Barman and Verma, 2026; Barman et al., 2025; Bogomolnaia et al., 2017; Chaudhury et al., 2023; Hosseini et al., 2023; Kulkarni et al., 2021a,b; Liu et al., 2024]. In this setting, an item can be a good that yields positive utilities to some agents and a chore that yields negative utilities to others. In order to circumvent the issue that an envy-free allocation may not exist with indivisible items, envy-freeness is often relaxed to *envy-freeness up to one item (EF1)* [Aziz et al., 2022; Budish, 2011; Lipton et al., 2004], a highly influential, robust, and widely studied concept. EF1 requires that any envy an agent has towards another agent vanishes if we ignore some chore in the former agent's bundle or some good in the latter agent's bundle. An EF1 allocation of mixed indivisible goods and chores always exists and can be computed efficiently for agents with additive or even doubly monotone utilities [Aziz et al., 2022; Bhaskar et al., 2021].

What if we are presented with both indivisible items and a cake? This model of mixed divisible and indivisible resources captures real-world applications such as rent division, divorce settlement, inheritance division, and any other situation involving both assets and liabilities to be divided between multiple entities. A compelling relaxation of envy-freeness in the mixed-resources model is *envy-freeness for mixed resources (EFM)* [Bei et al., 2021a; Bhaskar et al., 2021], which simultaneously combines the ideas of envy-freeness and EF1 in a meaningful and natural way. Phrased in our setting, EFM requires that if an agent i envies another agent j, then agent j should not get any sliver of the divisible resource for which agent i has positive utility, and moreover, agent i should be EF1 towards agent j. When all items are divisible, EFM coincides with envy-freeness; when the items are all indivisible, EFM coincides with EF1. With indivisible goods and a cake, an EFM allocation always exists for any number of agents with additive utilities [Bei et al., 2021a].

Bhaskar et al. [2021] revisited the work of Bei et al. [2021a] and examined the existence of EFM allocations in the presence of indivisible chores and a cake. Bhaskar et al. were only able to establish the existence of EFM allocations in two special cases: when agents have identical rankings over indivisible chores, and when the number of chores exceeds the number of agents by at most one. Bhaskar et al. [2021, p. 15] discussed in detail the technical challenges concerning the problem and emphasized that the solution will require fundamentally novel techniques. The problem has again been highlighted in the survey paper by Liu et al. [2024, Open Question 7]. The fundamental question that we explore in this paper and which has been raised in previous work [Bhaskar et al., 2021; Liu et al., 2024; Mishra et al., 2023] is as follows.

With indivisible items and a cake, does an EFM allocation always exist?

1.1 Our Contribution

We settle the above question in the affirmative by establishing the existence of EFM allocations in full generality for the case where there is a cake and m indivisible items (that may be subjective goods or chores) to be divided among n agents who have additive utilities.

Main Result 1 (Theorem 4.1). For indivisible items and cake, an EFM allocation always exists.

En route, we uncover an interesting connection among the solution concepts of envy-freeness, *envy-freeability*, EF1, and EFM, where envy-freeability is a concept developed and investigated in

the closely related but separate line of research on fair division with subsidy [e.g. Brustle et al., 2020; Halpern and Shah, 2019; Lu et al., 2026]. An allocation of indivisible items is said to be *envy-freeable* if the allocation can be made envy-free by allocating a sufficient amount of money to each of the agents [Halpern and Shah, 2019]. With only indivisible goods, Brustle et al. [2020] showed that an EF1 and envy-freeable allocation always exists.

We reduce the problem of showing the existence of an EFM allocation given any set of indivisible items and any cake to the problem of proving the existence of an EF1 and envy-freeable allocation of the indivisible items. We establish the following result, extending the result of Brustle et al. [2020] to the setting that includes both goods and chores.

Main Result 2 (Theorem 4.2). For indivisible goods and chores, an EF1 and envy-freeable allocation always exists. Moreover, such an allocation can be computed in polynomial time.

Our results significantly and simultaneously generalize several fundamental results in the literature, including the following:

- For cake, an envy-free allocation is guaranteed to exist [Alon, 1987; Su, 1999].
- For indivisible goods, an EF1 allocation exists [Caragiannis et al., 2019; Lipton et al., 2004].
- For indivisible goods and chores, an EF1 allocation exists [Aziz et al., 2022; Bhaskar et al., 2021].
- For indivisible goods, an EF1 and envy-freeable allocation exists and can be computed in polynomial time [Brustle et al., 2020].
- For indivisible goods and cake, an EFM allocation exists [Bei et al., 2021a].

1.2 Technical Challenges and Overview

As Bhaskar et al. [2021] pointed out, existing methods for obtaining EFM allocations fail in the setting that includes a mix of indivisible goods and chores due to the necessity of resolving *any* envy cycle, which may violate the EF1 property needed for EFM. We circumvent this issue by finding a fixed allocation of the indivisible items that will not be altered when we allocate the divisible good, while still achieving EFM. To achieve this, we first observe that the problem of allocating a heterogeneous cake and indivisible items is reducible to the problem of allocating indivisible items and money by showing that one can simulate money via cake. Hence, to achieve EFM, it suffices to find an indivisible-items allocation that is EF1 and can be made envy-free when there is a sufficient amount of money (i.e., an envy-freeable allocation). We establish this in Section 3.

Having reduced the problem of showing the existence of EFM to finding a fixed allocation of the indivisible items that is both EF1 and envy-freeable, we face the challenge of identifying such allocations, which presents its own technical difficulties. When there are only indivisible goods, Brustle et al. [2020] showed that such allocations exist. However, their algorithm violates EF1 in the setting with both goods and chores. On the other hand, all existing methods for obtaining EF1 allocations for goods and chores (such as those presented in [Aziz et al., 2022; Bhaskar et al., 2021]) violate envy-freeability. We overcome this by carefully bundling items together, treating them as *meta-items*, and then running an iterative matching algorithm similar to that of Brustle et al. [2020] in a way that yields both EF1 and envy-freeability.

The key technical contribution of our paper is to design polynomial-time algorithms, in the setting of both goods and chores, that return an EF1 and envy-freeable allocation by carefully

bundling items together. We achieve this by starting with an initial bundling of the items (Algorithm 2), which outputs a set of meta-items, each of which is considered good by some agent, along with the remaining individual objective chores. We then branch into two cases depending on the number of remaining individual chores: (i) the number of objective chores is at least n (see Section 4.1), and (ii) the number of objective chores is less than n (see Section 4.2).

In the first case, each meta-good can be combined with a distinct objective chore such that the resulting instance consists solely of objective chores. We construct the pairing and the allocation together, maximizing each round's matching value subject to the matchings already fixed and the feasibility of completing the pairing; a maximum-profit flow implements each round in polynomial time. We prove that this ensures EF1 and envy-freeability (Proposition 4.9).

In the second case, we further refine the bundling (Algorithm 6), which may partially undo the initial bundling, and consider different possible configurations of combining the resulting meta-items. This case fundamentally differs from the first one, as some agents may receive a meta-good. We prove that there exists a bundling that cleanly separates the agents, ensuring that if an agent receives a meta-good, then every other agent who receives a meta-chore perceives that agent's bundle as a meta-chore as well. To construct this bundling when at least one objective chore remains, we use a greedy expansion procedure (Algorithm 7) that jointly chooses a matching and one meta-chore to expand. Each iteration is implemented by a maximum-profit flow, and there are at most n expansions. The procedure returns the constructed meta-chores together with a maximum-weight matching that assigns them to distinct agents. An alternating-path argument ensures that each recipient values its attachment set non-negatively, even when later iterations reassign the meta-chores. We then allocate the remaining meta-goods and subjective goods to the remaining agents by iterative maximum-weight matching. Together, these steps yield an EF1 and envy-freeable allocation (Proposition 4.15).

1.3 Related Work

The fair division literature is vast and can be divided along at least two orthogonal directions: (i) goods versus chores and (ii) indivisible versus divisible items. We refer interested readers to the survey articles of Amanatidis et al. [2023], Guo et al. [2023], Nguyen and Rothe [2023], and Suksompong [2021, 2025] for an overview of the recent developments in the fair allocation of indivisible goods or indivisible chores. For settings with mixed indivisible goods and chores, or with mixed divisible and indivisible resources, we refer to the recent survey article of Liu et al. [2024] for an overview.

Our paper is mostly inspired by the work of Bei et al. [2021a], Bhaskar et al. [2021], and Brustle et al. [2020]. We will discuss in detail the recent developments in related lines of research.

Fair Division of Mixed Resources. This line of research focuses on the axiomatic study of fairness when allocating a mix of divisible and indivisible resources (i.e., *mixed resources*). Bei et al. [2021a] were the first to study an envy-freeness relaxation in the setting with only goods and introduced a version of EFM that is artificially stronger than our EFM definition.[1] Subsequent work has delved further into understanding EFM by investigating the design and limitations of *truthful* and EFM mechanisms in the presence of strategic agents [Li et al., 2023], quantitatively measuring the social welfare loss resulting from enforcing EFM through the lens of *price of*

[1] Bei et al. [2021a, Definition 2.3] requires envy-freeness from an agent i to another agent j as long as agent j's bundle contains a positive amount of divisible goods, even in the case that agent i values the divisible goods in j's bundle at 0. It is also known that the definition of Bei et al. is incompatible with Pareto optimality, a fundamental economic efficiency notion in the field. Our EFM definition is arguably more natural.

fairness [Li et al., 2024b], and exploring how to simultaneously satisfy exact fairness (e.g., envy-freeness and proportionality) ex ante while ensuring EFM ex post [Bu et al., 2024]. Other fairness notions, including the *maximin share (MMS) guarantee* [Bei et al., 2021b], relaxations of envy-freeness and proportionality based on *indivisibility ratio* [Li et al., 2024a], and a strengthening of EFM to EFXM [Nishimura and Sumita, 2025], have also been studied in the mixed-goods model.

Bhaskar et al. [2021] generalized the mixed-goods model and the EFM notion further to the mixed-resources model in which items could be goods or chores. In the setting with *doubly monotone* indivisible items and a divisible chore,[2] Bhaskar et al. showed an EFM allocation always exists.

Slightly further afield, Aziz et al. [2025] and Bei et al. [2025] relaxed the assumption that agents agree on whether each resource is divisible or indivisible.

Fair Division with Subsidy. The study of fair allocation with monetary transfers is well-established, particularly in the context of room-rent division. Most of the literature in this area assumes that each agent demands exactly one good or room [Aragones, 1995; Gal et al., 2017; Klijn, 2000; Svensson, 1983]. More general frameworks, where envy-freeness is achieved through side-payments, have been explored [Haake et al., 2002]. More recently, research has focused on computing envy-free allocations when agents may demand multiple goods and monetary transfers are permitted. In particular, Halpern and Shah [2019] brought attention to the problem of determining allocations that achieve envy-freeness through minimal subsidies. Subsequent work has delved deeper into the computational aspects of this problem, examining both exactly minimal subsidies [Brustle et al., 2020; Kawase et al., 2025] and approximately minimal subsidies [Caragiannis and Ioannidis, 2021], as well as its strategic aspects, examining fair and truthful mechanisms with limited subsidy [Goko et al., 2024].

While those works consider a goods-only setting, recently, Lu et al. [2026] studied envy-freeness with subsidies in the setting with mixed goods and chores. They showed that when the value of each item is bounded between -1 and 1, a subsidy of at most one dollar per agent suffices to guarantee an envy-free allocation. It is worth noting that the compatibility between EF1 and envy-freeability (i.e., our Main Result 2) does not necessarily lead to the desired subsidy bound. At the same time, the indivisible-items allocation obtained by Lu et al. [2026] to achieve the subsidy bound may not satisfy EF1.

2 Preliminaries

We study fair allocation of indivisible items (consisting of both goods and chores) and a divisible good among a set of agents $N = \{1, \dots, n\}$. For any positive integer k, let $[k] \coloneqq \{1, 2, \dots, k\}$.

Indivisible Items. Denote by $M = [m]$ the set of *indivisible items*. Each agent $i \in N$ has a utility function $u_i \colon 2^M \to \mathbb{R}$ that assigns a real value to each subset of items, with $u_i(\emptyset) = 0$ for all $i \in N$. Agents' utilities are assumed to be *additive*, i.e., $u_i(S) = \sum_{t \in S} u_i(t)$ for each bundle $S \subseteq M$. For convenience, we write $u_i(t)$ to denote $u_i(\{t\})$ for a single item $t \in M$.

An item $t \in M$ can be a *good* for one agent i (i.e., $u_i(t) \geq 0$) but a *chore* for another agent j (i.e., $u_j(t) < 0$). An item t is said to be a *subjective good* if $u_i(t) \geq 0$ for some $i \in N$. An item t is said to be an *objective good* (resp., *objective chore*) if $u_i(t) \geq 0$ (resp., $u_i(t) < 0$) for all agents $i \in N$.

[2] With doubly monotone indivisible items, each agent can partition the set of items into objective goods and objective chores.

We refer to a setting with agents N, indivisible items M, and utilities $(u_i)_{i\in N}$ as an *indivisible-items instance* (henceforth, *instance*), denoted as $\langle N, M, (u_i)_{i\in N}\rangle$.

Divisible Resource. We consider two types of divisible resources: cake (a heterogeneous divisible resource) and money (a homogeneous divisible resource). We now formally define each of them.

A *cake* C is a heterogeneous divisible good represented by the unit interval $C = [0, 1]$. Each agent $i \in N$ has a non-atomic *density function* $f_i \colon [0, 1] \to \mathbb{R}_{\geq 0}$ that describes their preferences over the cake. For any measurable subset $X \subseteq [0, 1]$, i's utility for X is given by $u_i(X) = \int_{x\in X} f_i(x)\ \mathrm{d}x$.

Money is a homogeneous divisible resource. If agent $i \in N$ receives $p_i \geq 0$ units of money, their utility for this money is $u_i(p_i) = p_i$. That is, all agents value money identically at one unit of utility per unit of money. Note that money can be seen as a special case of cake, where each agent's density function is identically one.

Precision and Input Representation. When discussing the computational aspects, we assume that $u_i(t)$ for each $i \in N, t \in M$ and $u_i(C)$ for each $i \in N$ are all rational numbers.

Allocations. An *indivisible-items allocation* $A = (A_1, \ldots, A_n)$ is a partition of M such that $A_i \cap A_j = \emptyset$ for all $i \neq j$ and $\bigcup_{i\in N} A_i = M$. Each agent i receives bundle A_i, which could be empty.

An allocation of mixed divisible and indivisible resources is a tuple $\mathcal{A} = \{(A_i, C_i)\}_{i\in N}$ where $(A_1, \ldots, A_n)$ is a partition of M and $(C_1, \ldots, C_n)$ is a partition of $[0, 1]$ into measurable sets. Agent i receives indivisible items A_i and a piece of cake C_i, with total utility $u_i(A_i, C_i) := u_i(A_i) + u_i(C_i)$.

When the divisible resource is money, we denote an allocation of indivisible items and money as $\mathcal{A} = \{(A_i, p_i)\}_{i\in N}$, where agent i receives indivisible items A_i and $p_i \geq 0$ units of money.

2.1 Fairness Notions

In this work, we are concerned with the problem of fairly dividing a set of indivisible items and a cake. We begin with classical fairness notions before introducing our main concepts.

Definition 2.1 (Envy-freeness)**.** With a mix of indivisible items M and a cake C, an allocation $\{(A_i, C_i)\}_{i\in N}$ is said to be *envy-free* if for every pair of agents $i, j \in N$, $u_i(A_i, C_i) \geq u_i(A_j, C_j)$.

When the set of indivisible items is empty and we only divide cake C, the problem is known as *cake cutting*. An envy-free cake cutting always exists [Alon, 1987; Su, 1999]. With only indivisible items, however, envy-freeness is a demanding property and cannot always be satisfied. It is thus often relaxed to *envy-freeness up to one item (EF1)*. An EF1 allocation of indivisible items always exists [Aziz et al., 2022; Bhaskar et al., 2021].

Definition 2.2 (EF1)**.** An indivisible-items allocation $(A_i)_{i\in N}$ is said to be *envy-free up to one item (EF1)* if for all $i, j \in N$,

- either $u_i(A_i) \geq u_i(A_j)$, or
- there exists an item $t \in A_i \cup A_j$ such that $u_i(A_i \setminus \{t\}) \geq u_i(A_j \setminus \{t\})$.

Fair Division of Mixed Divisible and Indivisible Resources. We are now ready to introduce our central fairness concept, which works for the setting with both divisible and indivisible resources and combines envy-freeness and EF1 naturally.

Definition 2.3 (EFM)**.** With indivisible items M and cake C, an allocation $\{(A_i, C_i)\}_{i \in N}$ is said to be *envy-free for mixed resources (EFM)* if for every pair of agents $i, j \in N$, either agent i is envy-free toward agent j, i.e., $u_i(A_i) + u_i(C_i) \geq u_i(A_j) + u_i(C_j)$, or both of the following conditions hold:

- Agent i does not value the cake allocated to agent j, i.e., $u_i(C_j) = 0$; and
- There exists an item $t \in A_i \cup A_j$ such that $u_i(A_i \setminus \{t\}) \geq u_i(A_j \setminus \{t\})$.

Remark. In the second bullet point of Definition 2.3, we could also include agent i's utility over her allocated divisible good in the definition; "weak EFM" of Bei et al. [2021a, Definition 6.4] is defined in this way. However, our Definition 2.3 is slightly stronger and can always be satisfied.

Since EFM is a comparison-based fairness notion, it is scale-invariant: multiplying an agent's utility function by any positive constant uniformly across both divisible and indivisible items does not affect EFM. Hence, without loss of generality, by applying such a uniform scaling, we may assume that for all agents, either $u_i(C) = 1$ or $u_i(C) = 0$. Throughout the paper, we adopt this normalization when studying the existence of EFM allocations.

When the EFM definition is applied to a homogeneous cake (money), an allocation of indivisible items and money, denoted as $\mathcal{A} = \{(A_i, p_i)\}_{i \in N}$, is said to be EFM if, for every pair of agents $i, j \in N$, either agent i is envy-free toward agent j, i.e., $u_i(A_i) + p_i \geq u_i(A_j) + p_j$, or both of the following conditions hold:

- agent j does not receive any money, i.e., $p_j = 0$; and
- there exists an item $t \in A_i \cup A_j$ such that $u_i(A_i \setminus \{t\}) \geq u_i(A_j \setminus \{t\})$.

One might wonder if we can further strengthen EFM by replacing the EF1 criterion with the stronger EFX criterion; see the corresponding definition below.

Definition 2.4 (EFXM)**.** With indivisible items M and cake C, an allocation $\{(A_i, C_i)\}_{i \in N}$ is said to be *envy-free up to any item for mixed resources (EFXM)* if for every pair of agents $i, j \in N$, either agent i is envy-free toward agent j, i.e., $u_i(A_i) + u_i(C_i) \geq u_i(A_j) + u_i(C_j)$, or all of the following conditions hold:

- There is no $C' \subseteq C_j$ such that $u_i(C') > 0$; and
- For each $t \in \{g \in A_j \mid u_i(g) > 0\} \cup \{c \in A_i \mid u_i(c) < 0\}$, it is the case that $u_i(A_i \setminus \{t\}) \geq u_i(A_j \setminus \{t\})$.

It would be interesting to explore to what extent an allocation satisfying EFXM is guaranteed to exist. For instance, with both divisible and indivisible goods, Nishimura and Sumita [2025] showed that for binary linear utilities, a maximum Nash welfare allocation satisfies EFXM.

In our setting with both goods and chores, however, He and Tao [2027] showed that an EFX allocation may not exist, even with only objective chores and under tri-valued additive utilities. When there is no cake, EFXM coincides with EFX; hence, by the result of He and Tao, we see that EFXM is not guaranteed to exist. In this sense, EFM is the strongest known concept for which a general existence result holds for agents with additive utilities in the presence of indivisible goods and chores.

Fair Division of Indivisible Items with Subsidy. Halpern and Shah [2019] formalized the concept of *envy-freeability* to capture allocations of indivisible goods that can be made envy-free through monetary compensation from a third-party. While their definition and characterization of envy-freeability were framed in the setting with only indivisible goods, they can be applied straightforwardly to the setting with mixed indivisible goods and chores.

Definition 2.5 (Envy-freeability)**.** An indivisible-items allocation $A = (A_1, \ldots, A_n)$ is said to be *envy-freeable* if there exists a vector of payments $(p_1, \ldots, p_n)$ such that $\{(A_i, p_i)\}_{i \in N}$ is envy-free.

Halpern and Shah [2019] gave a useful characterization of envy-freeability using the envy graph. Given an indivisible-items allocation $A = (A_i)_{i \in N}$ of M, its *envy graph* $\mathcal{G}_A$ is the complete weighted directed graph in which each agent is a node, and each edge (i, j) has weight $w_A(i, j) \coloneqq u_i(A_j) - u_i(A_i)$, i.e., the amount of envy that agent i has for agent j. The weight of any (directed) path P is defined as $w_A(P) = \sum_{(i,j) \in P} w_A(i, j)$, i.e., the sum of the weights of the edges along the path.

Theorem 2.6 (Halpern and Shah, 2019)**.** *For any allocation $A = (A_1, \ldots, A_n)$ of indivisible items, the following statements are equivalent:*

(i) Allocation A is envy-freeable.

(ii) Allocation A maximizes the social welfare across all reassignments of its bundles among the agents. That is, for every permutation σ over N, we have $\sum_{i=1}^{n} u_i(A_i) \geq \sum_{i=1}^{n} u_i(A_{\sigma(i)})$.

(iii) The envy graph $\mathcal{G}_A$ has no positive-weight cycles.

Note that an envy-freeable allocation need not be fair (e.g., EF1). It only requires the indivisible-items allocation combined with enough subsidy payments to be envy-free. Notably, with only indivisible goods, Brustle et al. [2020] show that an EF1 and envy-freeable allocation always exists. We establish below an analogous result in the setting with only indivisible chores.

Proposition 2.7. *With only indivisible chores, an EF1 and envy-freeable allocation always exists. Moreover, such an allocation can be found in polynomial time.*

At a high level, this result follows from first padding the instance with dummy chores so that the number of items is a multiple of n, and then running an *iterative maximum-weight perfect matching* on an appropriately defined graph. This graph has the property that each partial allocation obtained in each round is envy-freeable. Due to the additivity of the utilities, when adding up the partial allocations across all rounds, envy-freeability is preserved. By the nature of maximum-weight perfect matching, each agent prefers the item they are given in the current round to any item available in subsequent rounds. Finally, EF1 follows since each agent may remove their chore from the last round to achieve envy-freeness towards any other agent.

Proof of Proposition 2.7. We will show that Algorithm 1 outputs an allocation that is both EF1 and envy-freeable. Since there are exactly $T \cdot n$ chores in $\widehat{M}$, Algorithm 1 terminates in T rounds. Let $J = \widehat{M}$, and the graph $H[N, J]$ be a complete weighted bipartite graph. For each $i \in N$ and $c \in J$, edge (i, c) has weight $u_i(c)$. In each round t, each agent $i \in N$ is matched to a chore μ_i^t (which could be a dummy item). As a result, each agent $i \in N$ receives a bundle $A_i = \{\mu_i^1, \ldots, \mu_i^T\}$.

Allocation A is EF1. Observe that $u_i(\mu_i^t) \geq u_i(c)$ for any chore $c \in J_{t+1}$; otherwise, replacing the edge (i, μ_i^t) with (i, c) in μ^t results in a matching of strictly larger weight in $H[N, J_t]$. For any pair

Algorithm 1: Iterative Perfect Matching for Indivisible Chores

Input: Agents N, indivisible chores M with $|M| = T \cdot n - k$ and agents' utilities $(u_i)_{i \in N}$.
Output: An EF1 and envy-freeable allocation $A = (A_1, \ldots, A_n)$.

$A_i \leftarrow \emptyset$ for all $i \in N$.
Extend M to $\widehat{M}$ by adding k dummy items, each of which is valued at 0 for all agents.
$t \leftarrow 1$; $J_1 \leftarrow \widehat{M}$
while $J_t \neq \emptyset$ **do**
    Compute a maximum-weight perfect matching $\mu^t = \{(i, \mu_i^t)\}$ in $H[N, J_t]$.
    $A_i \leftarrow A_i \cup \{\mu_i^t\}$ for all $i \in N$.
    $J_{t+1} \leftarrow J_t \setminus \bigcup_{i \in N} \{\mu_i^t\}$
    $t \leftarrow t + 1$
return *allocation* $A = (A_1, A_2, \ldots, A_n)$

of agents $i, j \in N$, we have

$$\begin{aligned} u_i(A_i \setminus \{\mu_i^T\}) = u_i(\{\mu_i^1, \ldots, \mu_i^{T-1}\}) &= u_i(\mu_i^1) + \cdots + u_i(\mu_i^{T-1}) \\ &\geq u_i(\mu_j^2) + \cdots + u_i(\mu_j^T) \\ &\geq u_i(\mu_j^1) + u_i(\mu_j^2) + \cdots + u_i(\mu_j^T) \\ &= u_i(A_j), \end{aligned}$$

meaning that the output allocation A of Algorithm 1 is EF1.

Allocation A is envy-freeable. The argument follows exactly the proof of Lemma 3.1 in the paper by Brustle et al. [2020]. For the sake of being self-contained, we still provide the proof below. By Theorem 2.6, it suffices to show envy graph $\mathcal{G}_A$ has no positive-weight cycles. Take any directed cycle γ in the envy graph $\mathcal{G}_A$ and assume without loss of generality that γ involves a sequence of nodes/agents $1, 2, \ldots, k$ for some $k \geq 2$. We have

$$\begin{aligned} w_A(\gamma) = \sum_{(i,j) \in \gamma} w_A(i,j) = \sum_{(i,j) \in \gamma} \big(u_i(A_j) - u_i(A_i)\big) &= \sum_{(i,j) \in \gamma} \sum_{t=1}^{T} \Big(u_i(\mu_j^t) - u_i(\mu_i^t)\Big) \\ &= \sum_{(i,j) \in \gamma} \sum_{t=1}^{T} w_{\mu^t}(i,j) = \sum_{t=1}^{T} \sum_{(i,j) \in \gamma} w_{\mu^t}(i,j) \end{aligned}$$

Let π_γ be the permutation of N under which $\pi_\gamma(i) = i + 1$ for each $i \in [k-1]$, $\pi_\gamma(k) = 1$, and $\pi_\gamma(i) = i$ for $i \in N \setminus [k]$. In each round t, μ^t is a maximum-weight matching; therefore, $\sum_{(i,j) \in \gamma} w_{\mu^t}(i,j) \leq 0$. Otherwise, giving each i the item $\mu^t_{\pi_\gamma(i)}$ results in a matching of strictly larger weight, a contradiction. Thus, $w_A(\gamma) \leq 0$, implying that allocation A is envy-freeable.

Polynomial-Time Computation. Let $m = |M|$. If $m = 0$, the empty allocation suffices. Otherwise, choose $T = \lceil m/n \rceil$ and $k = Tn - m$, so that $0 \leq k < n$ and the padded instance contains $Tn < m + n$ items. There are exactly $T \leq m$ matching rounds. In each round, a maximum-weight matching covering all n agents can be computed in polynomial time on a complete bipartite graph with at most $m + 2n$ vertices and at most $n(m + n)$ edges. For rational input utilities encoded in binary, all edge weights are either input utilities or zero, and hence have polynomial bit length. Using exact rational arithmetic, the matching computations and the allocation updates therefore take time polynomial in the input bit length. □

It is obvious that the goal of the subsidy literature is different from the axiomatic study of fairness for fair division with mixed divisible and indivisible resources. Nevertheless, in the next section, we will show that the tools developed in the subsidy literature are very useful to prove the existence of EFM allocations when we divide a fixed set of divisible and indivisible resources.

3 Cake to Money Reduction

In this section, we show that, when considering the existence of EFM allocations, the problem of allocating a cake among agents with heterogeneous valuations can be reduced to the problem of allocating money. A defining characteristic of money is that all agents value it equally; a dollar holds the same worth regardless of who possesses it. In contrast, when agents value the cake heterogeneously, the values assigned to different portions of the cake may vary arbitrarily across agents. Despite these differences, the following result demonstrates that, with appropriate scaling, it is possible to "simulate" money using a cake.

Theorem 3.1 (Stromquist and Woodall, 1985)**.** *Let $n \geq 2$, and let $(u_i)_{i \in N}$ be utility functions defined over the cake $C = [0,1]$, with $u_i(C) = 1$ for all $i \in N$. Then, for any $\alpha \in [0,1]$, there exists a set $I_\alpha \subseteq C$ such that I_α is the union of at most n intervals and satisfies $u_i(I_\alpha) = \alpha$ for every $i \in N$.*

The above result enables us to apply techniques from the fair division with subsidies literature to establish the existence of EFM allocations. We make this connection more precise below.

Theorem 3.2. *In the mixed indivisible goods and chores setting, if there exists an allocation of the indivisible items that is both EF1 and envy-freeable, then an EFM allocation exists with cake.*

Proof. Let $A = (A_1, \ldots, A_n)$ be an indivisible-items allocation that is both envy-freeable and EF1. We will show that there exists an allocation of the cake that preserves the underlying discrete allocation A and yields a combined allocation $\mathcal{A} = \{(A_i, C_i)\}_{i \in N}$ of indivisible items and cake that satisfies EFM. Recall that under the scaling, each agent values the entire cake either at 1 or 0. Let $N^+ := \{i \in N \mid u_i(C) = 1\}$ and $N^0 := N \setminus N^+$; note that for every $j \in N^0$, we have $u_j(C) = 0$.

Claim 3.3. *If there exists an EFM allocation with money, $\{(A_i, p_i)\}_{i \in N^+}$, among the agents in N^+ such that $\sum_{i \in N^+} p_i = 1$, then there exists a combined allocation $\mathcal{A} = \{(A_i, C_i)\}_{i \in N}$ that is EFM.*

We make use of the result of Stromquist and Woodall [1985] to prove the claim.

Proof of Claim 3.3. For each $p_i > 0$ with $i \in N^+$, we iteratively apply Theorem 3.1 to the cake C to generate pieces C_i's such that $u_j(C_i) = p_i$ for all $j \in N^+$. If at the end of this process, the cake C is not fully allocated, then the remaining cake must be valued at zero by every agent $i \in N^+$,

$$u_i\left(C \setminus \bigcup_{j \in N^+ \colon p_j > 0} C_j\right) = 1 - \sum_{j \in N^+ \colon p_j > 0} p_j = 0.$$

We allocate the remaining cake to any agent in N^+. At this stage, the entire cake is allocated. Consider now the full allocation of the indivisible items and the cake:

$$\mathcal{A} = \begin{cases} (A_i, C_i) & \text{if } i \in N^+; \\ (A_i, \emptyset) & \text{if } i \in N^0. \end{cases}$$

We now proceed to show allocation $\mathcal{A}$ satisfies EFM. Recall that the underlying indivisible-items allocation A is EF1. For any agent $j \in N^0$, since they value the entire cake at zero, any

measurable subset of the cake is also valued at zero. Thus, for these agents, EFM reduces to requiring that the allocation of indivisible items be EF1, a condition that the discrete allocation A satisfies.

For any agent $j \in N^+$, if they envy another agent $k \in N^0$, then EFM holds because agent k does not receive any cake and the underlying discrete allocation A is EF1.

Finally, for any pair of agents $i, j \in N^+$, if i is envy-free toward j in the allocation $\{(A_i, p_i)\}_{i \in N^+}$, then the envy-free relation holds in $\mathcal{A}$ since $u_i(A_i) + u_i(C_i) = u_i(A_i) + p_i \geq u_i(A_j) + p_j = u_i(A_j) + u_i(C_j)$. On the other hand, if i envies j in $\{(A_i, p_i)\}_{i \in N^+}$, then $p_j = 0$. Since the underlying discrete allocation A is EF1, the EFM condition is satisfied for $\mathcal{A}$. □

Claim 3.3 establishes that, for an EF1 and envy-freeable allocation A, it suffices to find payments among agents N^+ that make them EFM with money. We now show how to construct such payments.

Note that the allocation A restricted to N^+ is envy-freeable: any positive-weight cycle among the agents in N^+ would also be a positive-weight cycle in the full envy graph $\mathcal{G}_A$, contradicting the envy-freeable property of A. Thus, no such cycle exists.

Let $\widetilde{\mathcal{G}}$ be the envy-graph of allocation A restricted to agents N^+, i.e., the subgraph of $\mathcal{G}_A$ induced by the vertex set N^+. For each $i \in N^+$, let $\ell(i)$ denote the weight of the heaviest path starting from node i in $\widetilde{\mathcal{G}}$. Since $\widetilde{\mathcal{G}}$ has no positive-weight cycles, $\ell(i)$ is well-defined. By Theorem 2 of Halpern and Shah [2019], assigning each agent $i \in N^+$ a payment $q_i = \ell(i)$ yields an envy-free allocation with money among agents in N^+.

If $\sum_{i \in N^+} q_i \leq 1$, set $p_i = q_i + \frac{1 - \sum_{i \in N^+} q_i}{|N^+|}$ for each $i \in N^+$. Then $\{(A_i, p_i)\}_{i \in N^+}$ remains envy-free and satisfies $\sum_{i \in N^+} p_i = 1$, establishing EFM.

If instead $\sum_{i \in N^+} q_i > 1$, we distribute the unit amount of money while preserving EFM. Let $q^{(1)} > q^{(2)} > \cdots > q^{(k)} = 0$ denote the *distinct* payment values among the $(q_i)_{i \in N^+}$, and for each t, let $N^t = \{i \in N^+ \mid q_i = q^{(t)}\}$. We proceed iteratively to build up the desired payments:

- In round 1, allocate an amount of money worth $q^{(1)} - q^{(2)}$ to each agent in N^1.
- In round 2, allocate an amount of money worth $q^{(2)} - q^{(3)}$ to each agent in $N^1 \cup N^2$.
- Continue similarly, giving in round r an amount $q^{(r)} - q^{(r+1)}$ to each agent in $\bigcup_{t=1}^{r} N^t$, until the total allocated amount reaches exactly 1.

  If, in some round r, the remaining money is insufficient to allocate the full increment $q^{(r)} - q^{(r+1)}$, we instead divide the remaining money equally among agents in $\bigcup_{t=1}^{r} N^t$; denote by $\widehat{q}$ the amount each agent receives. Clearly, the total money distributed equals 1.

Let $\{(A_i, p_i)\}_{i \in N^+}$ be the resulting allocation with payments. By construction, we have that $\sum_{i \in N^+} p_i = 1$. We now show $\{(A_i, p_i)\}_{i \in N^+}$ satisfies EFM with money. If $p_i, p_j > 0$, then $p_i - p_j = q_i - q_j$ by construction, and hence i and j are envy-free towards each other since the allocation $\{(A_i, q_i)\}_{i \in N^+}$ is envy-free. Suppose $p_j > 0$ and $p_i = 0$; we must show that i is envy-free towards j. Since $p_j > 0$ and $p_i = 0$, we have $q_j > q_i$, meaning $j \in N^a$ and $i \in N^b$ with $a \leq r < b$, where r is the final round in which money is allocated. Then,

$$q_j - q_i = q^{(a)} - q^{(b)} = \sum_{t=a}^{b-1} \left(q^{(t)} - q^{(t+1)} \right) \geq \sum_{t=a}^{r-1} \left(q^{(t)} - q^{(t+1)} \right) + \widehat{q} = p_j.$$

Recall that $\{(A_i, q_i)\}_{i \in N^+}$ is envy-free; thus, we have $u_i(A_i) + q_i \geq u_i(A_j) + q_j$. Agent i is envy-free towards j because $u_i(A_i) + p_i = u_i(A_i) \geq u_i(A_j) + (q_j - q_i) \geq u_i(A_j) + p_j$. Since the discrete

allocation A is EF1 and no agent envies another who received a positive payment, $\{(A_i, p_i)\}_{i \in N^+}$ satisfies EFM with money. Applying Claim 3.3, we obtain an EFM allocation with cake. □

Theorem 3.2 establishes an interesting connection between solution concepts, i.e., envy-freeness, EF1, envy-freeability, and EFM, investigated in two closely related but separate lines of research.

With only indivisible goods, an EF1 and envy-freeable allocation always exists [Brustle et al., 2020]. Therefore, the proof of Theorem 3.2 provides a novel approach to compute an EFM allocation for agents with additive preferences over a set of indivisible goods and a cake. Our method exhibits different properties compared with the algorithm devised by Bei et al. [2021a].

On the one hand, Bei et al.'s algorithm indeed produces an allocation satisfying a stronger envy-freeness relaxation and is able to handle monotone utilities over the indivisible goods [see, Liu et al., 2024, pp. 1389 ff.]. On the other hand, however, as Bhaskar et al. [2021, pp. 15] pointed out, Bei et al.'s framework fails in the setting involving indivisible chores and a cake. In more detail, Bei et al.'s algorithm relies crucially on resolving envy cycles among agents so that the algorithm can continue allocating more cake to the current allocation. The challenge is that with indivisible chores, we can no longer resolve *any* envy cycle.

Our method presented in the proof of Theorem 3.2 *preserves* the underlying discrete allocation. Put differently, we are able to hand out pieces of cake to agents without permuting agents' bundles. This property makes our method naturally fit to tackle the said challenge, even in the more general setting with both indivisible goods and chores, as long as an EF1 and envy-freeable indivisible-items allocation always exists.[3]

4 Existence of EFM

We are now ready to present the main results of our paper.

Theorem 4.1. *For indivisible items and cake, an EFM allocation always exists.*

We will prove Theorem 4.1 by showing that given any instance of indivisible goods and chores, an EF1 and envy-freeable allocation always exists, and hence by Theorem 3.2, the existence of an EFM allocation is guaranteed.

Theorem 4.2. *For indivisible goods and chores, an EF1 and envy-freeable allocation always exists. Moreover, such an allocation can be computed in polynomial time.*

Although we can compute an EF1 and envy-freeable allocation in polynomial time, we would like to remark that we only focus on the existence result of EFM. This is because our cake to money reduction described in Section 3 does not admit a finite protocol under the Robertson-Webb model and our algorithm uses it as a subroutine.[4]

The remainder of this section is dedicated to proving Theorem 4.2, which in and of itself is of independent interest.

[3] Combining Proposition 2.7 and Theorem 3.2, it follows immediately that with indivisible chores and a cake, an EFM allocation always exists.

[4] Theorem 3.1 provides a cake division that is stronger than perfect division, which cannot be implemented with a finite number of queries in the Robertson-Webb model, even if there are only two agents [Robertson and Webb, 1998].

Bundling Plus Iterative Matching

We first explain the high-level approach behind Theorem 4.2. As the technique of applying iterative matching has obtained positive results in settings with only goods [Brustle et al., 2020] or chores (Proposition 2.7), it is therefore the immediate candidate tool for us to show EF1 and envy-freeability in the presence of mixed goods and chores. The idea of applying iterative matching, however, fails without bundling items together. This can be seen from a simple instance with two items and two agents who have an identical utility over the two items: one item is of value $+1$ while the other item is of value -1. The only way to achieve both EF1 and envy-freeability is to bundle the two items together and allocate the bundle to any agent.

The main idea of the proof is to carefully *bundle* certain items together and treat each resulting bundle as a single *meta-item*. It is also key to us that the way we bundle items should be consistent with how the iterative matching algorithm allocates meta-items to the agents because both EF1 and envy-freeability will be ensured by the property of iterative matching.

In more detail, this bundling step yields a collection of pairwise disjoint sets $J = \{S_1, \ldots, S_r\}$, where each $S_i \subseteq M$. After bundling, we construct two bipartite graphs whose bipartitions correspond to the agent set N and the bundled item set J (each element $S \in J$ represents a node):

- bipartite graph $H[N, J]$, designed to handle objective chores (bundled or otherwise), and
- bipartite graph $G[N, J]$, designed to handle subjective goods (also bundled or otherwise).

For each edge (i, S), where $S \in J$ corresponds to a bundle, the edge weight is given by $u_i(S)$. The graph structures of G and H differ slightly to capture the distinct nature of goods and chores. Finally, we run the *Iterative Maximum-Weight Matching* (`IMWM`) algorithm on G and the *Iterative Maximum-Weight Perfect Matching* (`IMWPM`) algorithm on H to determine the final allocation. The key challenge is to choose the bundling of items such that running the above algorithms yields an allocation that is both envy-freeable and EF1.

We now define several useful terms.

Definition 4.3 (Meta-good and interest set)**.** We refer to a non-empty subset $M' \subseteq M$ of indivisible items as a *meta-good* if $u_i(M') \geq 0$ for some agent $i \in N$.

Moreover, we will refer to the set of agents who derive non-negative utility from the meta-good M', i.e., $\{i \in N \mid u_i(M') \geq 0\}$, as its *interest set*.

Note that a meta-good may be a singleton. In our algorithms and arguments, we treat each meta-good as an *individual* item rather than as a set of items. A natural restriction on meta-goods, when reasoning about EF1, is the following property.

Definition 4.4 (Chore-maximality)**.** A non-empty subset $M' \subseteq M$ of indivisible items is said to be *chore-maximal* if $u_i(M') \geq 0$ for some $i \in N$ (i.e., M' is a meta-good) and for any objective chore $c \in M \setminus M'$, we have $u_j(M' \cup \{c\}) < 0$ for all $j \in N$.

Initial Bundling

We begin with an initial bundling that forms the foundation for subsequent bundling steps, depending on the structure of the instance. Broadly speaking, this initial bundling guides later bundling operations and the branching of the instance into different cases.

Our initial bundling of the items aims to partition M into a set of objective chores and meta-goods each of which satisfies chore-maximality. We additionally require that each agent values at most one meta-good non-negatively. This is achieved by Algorithm 2.

Algorithm 2: Iterative Item Merging

Input: Agents N, indivisible items M, and agents' utilities $(u_i)_{i \in N}$.
Output: A set of meta-goods $\{M_1, \ldots, M_\ell\}$ and a set of objective chores Z.

1 Let $U \subseteq M$ be the set of subjective goods in M and $Z = M \setminus U$ be the set of objective chores.
2 **while** *there exists an agent* $i \in N$ *with* $\deg(i) \geq 2$ *in* $R[N, U]$ **do**
3 Select an arbitrary agent $i \in N$ with $\deg(i) \geq 2$.
4 Let $S = \{g \in U \mid (i, g) \in E\}$. `// Neighbourhood items of` i.
5 Create a new node S^* representing S. `// Merge neighbourhood items` S `into a meta-good.`
6 **foreach** *agent* $j \in N$ **do** Add an edge (j, S^*) to E if $u_j(S^*) \geq 0$.
7 Remove all edges incident to items in S from E.
8 Update item set $U \leftarrow (U \setminus S) \cup \{S^*\}$.
9 Let $\{M_1, \ldots, M_\ell\}$ be the set of ℓ meta-goods in $R[N, U]$. `// Each node in` U `is a meta-good.`
10 **while** $|Z| > 0$ **and** $\exists\, j \in [\ell], i \in N, c \in Z$ *such that* $u_i(M_j \cup \{c\}) \geq 0$ **do** `// Chore maximality.`
11 $M_j \leftarrow M_j \cup \{c\}$
12 $Z \leftarrow Z \setminus \{c\}$
13 **return** *A set of meta-goods* $\{M_1, \ldots, M_\ell\}$ *and objective chores* Z

Algorithm 2 takes as input an instance $\langle N, M, (u_i)_{i \in N} \rangle$ and iteratively merges items into meta-goods followed by adding chores appropriately to satisfy chore-maximality. Let $U \subseteq M$ be the set containing all subjective goods in M and let $Z = M \setminus U$ be the set of (current) objective chores. Consider the weighted bipartite graph $R[N, U] = (N \cup U, E)$, where there is an edge $(i, g) \in E$ for some $i \in N$ and $g \in U$ if $u_i(g) \geq 0$, with edge weight set to $u_i(g)$. The first `while`-loop (lines 2 to 8) of Algorithm 2 identifies an agent $i \in N$ whose degree is at least 2 in $R[N, U]$ and merges agent i's neighbourhood items $S = \{g \in J \mid (i, g) \in E\}$ into a meta-good S^*. We then update the bipartite graph by removing S from the item set U and all edges incident to items in S from E as well as by adding the new node S^* to the item set U and edges (j, S^*) to E if $u_j(S^*) \geq 0$. The `while`-loop terminates when each agent values at most one node in U non-negatively. In each iteration, $|U|$ strictly decreases; hence, the `while`-loop will terminate.

When the first `while`-loop (lines 2 to 8) terminates, assume that we have ℓ meta-goods, denoted as $M_1, M_2, \ldots, M_\ell$. Note that a meta-good $M_j \in \{M_1, M_2, \ldots, M_\ell\}$ may be a singleton. Recall that Z contains the set of objective chores. The second `while`-loop (lines 10 to 12) of Algorithm 2 further merges objective chores to the ℓ meta-goods so that each updated meta-good could also satisfy chore-maximality. In more detail, in each iteration of lines 10 to 12, we identify a meta-good M_j and an objective chore $c \in Z$ such that $u_i(M_j \cup \{c\}) \geq 0$ for some $i \in N$, merge c into M_j, and remove c from Z. Clearly, the updated M_j remains a meta-good and since we added an objective chore, we maintain the invariant that each agent values at most one meta-good non-negatively. Since $|Z|$ strictly decreases in each iteration, the `while`-loop will terminate. At its termination, either $|Z| = 0$ or all meta-goods $M_j \in \{M_1, \ldots, M_\ell\}$ satisfy chore-maximality. Recall from Definition 4.3: for any meta-good M_j, its *interest set*, denoted as $T_j := \{i \in N \mid u_i(M_j) \geq 0\}$, is the set of agents who derive non-negative utility from it. To summarize, we have the following lemma.

Lemma 4.5. *Given an instance $\langle N, M, (u_i)_{i\in N}\rangle$, let $\{M_1, \ldots, M_\ell\}$ be the set of meta-goods outputted by Algorithm 2. We have the following two properties.*

- *When $|Z| > 0$, each meta-good $M_j \in \{M_1, \ldots, M_\ell\}$ is chore-maximal.*
- *The interest sets are pairwise disjoint, i.e., $T_j \cap T_k = \emptyset$ for all $j \neq k$.*

Algorithm 2 runs in polynomial time.

Given any instance $\langle N, M, (u_i)_{i\in N}\rangle$, we apply Algorithm 2 and let $\{M_r\}_{r\in[\ell]}$ denote the set of meta-goods produced. We distinguish our analyses based on the number of remaining objective chores, defined as $Z = M \setminus \bigcup_{r=1}^{\ell} M_r$. The arguments required to establish the existence of EF1 and envy-freeable allocations differ depending on whether $|Z| \geq n$ or $|Z| \leq n-1$. At a high level, these case distinctions are necessary for several reasons. When $|Z| \geq n$, we can always combine each meta-good with a distinct objective chore from Z such that the resulting instance consists solely of objective chores. In contrast, when $|Z| \leq n-1$, no such combination may be possible, which introduces additional technical complications and necessitates a different bundling strategy. In the following subsections, we analyze these two cases in Sections 4.1 and 4.2, respectively.

4.1 Case I: At Least n Remaining Objective Chores ($|Z| \geq n$)

In this case, we have a collection of meta-goods $\{M_r\}_{r\in[\ell]}$ and a set of objective chores $Z = \{c_1, \ldots, c_k\}$ with $k \geq n$ outputted by Algorithm 2. Recall that each of the meta-goods in $\{M_r\}_{r\in[\ell]}$ is chore-maximal by Lemma 4.5. Hence, for any objective chore $c \in Z$, the combined bundle $M_r \circ c$ constitutes a meta-chore, i.e., $u_i(M_r \circ c) < 0$ for all $i \in N$.[5] Furthermore, by Lemma 4.5, T_j's are pairwise disjoint. Together with $\bigcup_{j\in[\ell]} T_j \subseteq N$, it follows that $\ell \leq n \leq k$. It means that each meta-good can be paired with a distinct objective chore to form a meta-chore.

Pairing Configuration. Let $\mathrm{Inj}([\ell], [k])$ denote the set of all injective functions $\varphi\colon [\ell] \to [k]$ (i.e., one-to-one assignments of the ℓ meta-goods $\{M_r\}_{r\in[\ell]}$ to ℓ distinct chores in Z). For each $\varphi \in \mathrm{Inj}([\ell], [k])$, define the corresponding set of meta-chores, which forms a partition of the item set M, as

$$J^\varphi = \left(\bigcup_{i\in[\ell]} M_i \circ c_{\varphi(i)}\right) \cup \left(Z \setminus \bigcup_{j\in[\ell]} c_{\varphi(j)}\right).$$

The first term represents the ℓ meta-chores obtained by combining each of the ℓ meta-goods with a distinct objective chore, while the second term consists of the remaining $k-\ell$ unattached chores that are singletons. Note that for any $\varphi \in \mathrm{Inj}([\ell], [k])$, the cardinality satisfies $|J^\varphi| = |Z|$.

We now introduce dummy items so that the total number of the extended set of items is a multiple of n. Let $T = \lceil k/n \rceil$ and $r = T \cdot n - k$. We introduce r dummy items $d_1, \ldots, d_r$, each valued at 0 by every agent. For each $\varphi \in \mathrm{Inj}([\ell], [k])$, we construct the padded instance with dummy items,

$$\widetilde{J}^\varphi = \left(\bigcup_{a\in[r]} d_a\right) \cup \left(\bigcup_{i\in[\ell]} M_i \circ c_{\varphi(i)}\right) \cup \left(Z \setminus \bigcup_{j\in[\ell]} c_{\varphi(j)}\right).$$

[5] We use the notation "$M_r \circ c$" to denote the operation that a meta-good M_r is *attached* to an objective chore c, and refer to $M_r \circ c$ as a *meta-chore* in order to distinguish it from a singleton chore. In our argument presented later, given a meta-chore $M_r \circ c$, we may *detach* the meta-good M_r from the singleton chore c. We thus decide to use a different notation for a meta-chore from that for a meta-good.

Algorithm 3: Iterative Maximum-Weight Perfect Matching (IMWPM)

Input: Graph $H[N, \widetilde{J}^{\varphi}]$ with $|\widetilde{J}^{\varphi}| = T \cdot n$
Output: An allocation $A = (A_1, \ldots, A_n)$.

$A_i \leftarrow \emptyset$ for all $i \in N$.
$t \leftarrow 1$; $J_1 \leftarrow \widetilde{J}^{\varphi}$
while $J_t \neq \emptyset$ **do**
  Compute a maximum-weight perfect matching $\mu^t = \{(i, \mu_i^t)\}$ in $H[N, J_t]$.
  $A_i \leftarrow A_i \cup \{\mu_i^t\}$ for all $i \in N$.
  $J_{t+1} \leftarrow J_t \setminus \bigcup_{i \in N} \{\mu_i^t\}$
  $t \leftarrow t + 1$
return *allocation* A

By construction, we have $|\widetilde{J}^{\varphi}| = T \cdot n$.

Roundwise Optimal Pairing. For a fixed pairing $\varphi \in \mathrm{Inj}([\ell], [k])$, let $H[N, \widetilde{J}^{\varphi}] = (N \cup \widetilde{J}^{\varphi}, E)$ be the complete bipartite graph between the agents and the objects. Here, each $h \in \widetilde{J}^{\varphi}$ is either a dummy item, a meta-chore of the form $M_r \circ c_{\varphi(r)}$ for some $r \in [\ell]$, or a singleton chore. For each $i \in N$ and $h \in \widetilde{J}^{\varphi}$, edge (i, h) has weight $u_i(h)$. For $\widehat{J} \subseteq \widetilde{J}^{\varphi}$, write $H[N, \widehat{J}]$ for the induced subgraph.

Given a fixed pairing, Algorithm 3 assigns one object to each agent in every round using a maximum-weight perfect matching among the remaining objects. Let $\mu^t = \{(i, \mu_i^t) : i \in N\}$ denote the matching in round t, and let $\mathrm{val}(\mu^t) = \sum_{i \in N} u_i(\mu_i^t)$.

We will construct the pairing and the matchings simultaneously. At each round, we maximize $\mathrm{val}(\mu^t)$ over assignments consistent with the previously fixed matchings and attachments and extendable to a complete pairing. We call this choice *roundwise optimal*; ties in the current round may be broken arbitrarily before proceeding to the next round. The resulting pairing is denoted by φ^*. Lemma 4.7 will show that the recorded rounds are also an execution of IMWPM for this final pairing.

Polynomial-Time Implementation of the Roundwise Optimal Pairing. We implement this sequential choice using a maximum-profit flow network, adapting the construction of Lu et al. [2026].

Let $Z_t \subseteq Z$ be the unallocated singleton chores and $B_t \subseteq [\ell]$ the indices of the unattached meta-goods at the beginning of round t. Initially, $Z_1 = Z$ and $B_1 = [\ell]$. We maintain $|Z_t| \geq |B_t|$ and set

$$q_t = \begin{cases} n - r, & t = 1, \\ n, & t > 1, \end{cases} \qquad b_t = |Z_t| - |B_t|. \tag{1}$$

Here, q_t is the number of real chores selected in round t. The algorithm allocates all r dummy items in the first round by construction, assigning a real object to each of the other $n - r$ agents. Every subsequent round assigns a real object to every agent. This convention is consistent with optimizing the current round: under any complete pairing, replacing a selected real object with an unused dummy strictly increases the matching value. A *candidate round* assigns q_t distinct chores to distinct agents and attaches p distinct meta-goods to distinct selected chores. When

$$|Z_t| - q_t \geq |B_t| - p, \qquad \text{equivalently,} \qquad q_t - p \leq b_t, \tag{2}$$

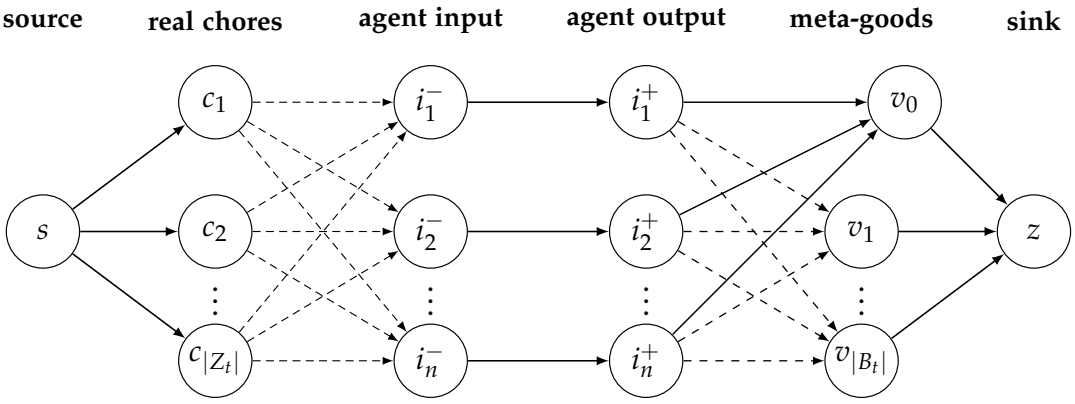


Figure 1: The maximum-profit flow network for round t. Flow through a meta-good node v pairs the selected chore with that meta-good; flow through v_0 leaves the selected chore unpaired.

it leaves enough singleton chores to attach all remaining meta-goods, meaning that this round is *extendable*.

For each round, construct a directed network F_t with source s, sink z, one node for each $c \in Z_t$, two nodes i^-, i^+ for each $i \in N$, one node v_j for each $j \in B_t$, and an auxiliary node v_0; see Figure 1. Its arcs, capacities, and profits are as follows; the indicated indices range over $c \in Z_t$, $i \in N$, and $j \in B_t$.

Arc	Capacity	Profit	Arc	Capacity	Profit
$s \longrightarrow c$	1	0	$i^+ \longrightarrow v_j$	1	$u_i(M_j)$
$c \longrightarrow i^-$	1	$u_i(c)$	$i^+ \longrightarrow v_0$	1	0
$i^- \longrightarrow i^+$	1	0	$v_j \longrightarrow z$	1	0
			$v_0 \longrightarrow z$	b_t	0

Given F_t, we compute an integral maximum-profit flow of *exactly* q_t units. A flow path through chore c, agent nodes i^-, i^+, and v_j assigns $M_j \circ c$ to agent i; a path using v_0 instead assigns only c to i. An agent carrying no flow receives a distinct dummy item. This last situation occurs only in the first round. The pseudocode can be found in Algorithm 4.

Lemma 4.6. *Integral flows of value q_t in F_t encode precisely the candidate rounds satisfying Equation (2), up to the identities of dummy items. The flow profit equals the matching value of the encoded round.*

Proof. The unit capacities ensure that each selected chore and each attached meta-good is used once, and that each agent receives at most one real object. If p meta-goods are attached, exactly $q_t - p$ units pass through v_0. The capacity of $v_0 \longrightarrow z$ therefore imposes Equation (2). Conversely, any such candidate round gives a feasible flow by routing each assigned object along its corresponding path. Since the network is acyclic, every integral flow decomposes into these paths. Their profits are $u_i(c)$ and $u_i(c) + u_i(M_j)$, respectively, exactly the utilities of the assigned objects. □

In the remainder of this subsection, we will show that Algorithm 4 returns an allocation that is both EF1 and envy-freeable. We first verify that every round is feasible and that the algorithm allocates all original items. Initially, $|Z_1| = k \geq \ell = |B_1|$ and $1 \leq q_1 = n - r \leq \min\{n, k\}$. Suppose that a round begins with $|Z_t| \geq |B_t|$ and $1 \leq q_t \leq \min\{n, |Z_t|\}$. Select any q_t chores and attach $p = \max\{0, q_t - b_t\}$ distinct meta-goods to distinct selected chores. Since $q_t \leq |Z_t|$ and

Algorithm 4: Case I: Roundwise Optimal Pairing and Allocation

Input: Meta-goods $M_1, \dots, M_\ell$ and chores Z from Algorithm 2, with $k = |Z| \geq n$.
Output: An EF1 and envy-freeable allocation of the original items.

$T \leftarrow \lceil k/n \rceil$; $r \leftarrow T \cdot n - k$; create r zero-valued dummy items.
$Z_1 \leftarrow Z$; $B_1 \leftarrow [\ell]$; $A_i \leftarrow \emptyset$ for every $i \in N$.
for $t = 1, \dots, T$ **do**
  Set $b_t = |Z_t| - |B_t|$ and $q_t = \begin{cases} n - r, & t = 1, \\ n, & t > 1. \end{cases}$
  Build F_t (see Figure 1) and compute an integral maximum-profit flow of value q_t.
  Decode the flow into objects μ_i^t, assigning a distinct unused dummy item to every agent carrying no flow. Add the original items represented by μ_i^t to A_i for every $i \in N$; ignore dummy items.
  Remove selected chores from Z_t and attached meta-good indices from B_t to obtain Z_{t+1} and B_{t+1}.
return $A = (A_1, \dots, A_n)$

$b_t = |Z_t| - |B_t| \geq 0$, we have $p \leq \min\{q_t, |B_t|\}$. Assigning the resulting objects to distinct agents gives an extendable candidate round and hence, by Lemma 4.6, a feasible flow of value q_t.

For the round selected by the algorithm, Equation (2) gives $|Z_{t+1}| \geq |B_{t+1}|$. The first round leaves $(T-1)n$ real chores, and each subsequent round removes exactly n. Thus, inductively, $|Z_t| = (T - t + 1)n \geq n = q_t$ for every $2 \leq t \leq T$, so all T rounds are feasible. At their completion, $Z_{T+1} = \emptyset$ and the invariant implies $B_{T+1} = \emptyset$. Each chore has been assigned once and each meta-good attached once, so all original items have been allocated. The recorded attachments define an injective map φ^*.

Lemma 4.7. *Let φ^* be the pairing produced by Algorithm 4, and let $(\mu^1, \dots, \mu^T)$ be its recorded matchings, including the assignments of dummy items. For each round t, define*

$$R_t = \widetilde{J}^{\varphi^*} \setminus \{\mu_i^s : i \in N,\ 1 \leq s < t\},$$

which contains the objects under this fixed pairing that remain after the algorithm's first $t-1$ rounds. Then μ^t is a maximum-weight perfect matching in $H[N, R_t]$. Consequently, the recorded sequence is an execution of IMWPM$(H[N, \widetilde{J}^{\varphi^*}])$*, with suitable tie-breaking.*

Proof. Fix a round t and the final pairing φ^*. Every chore in Z_t underlies exactly one real object in R_t. In the final pairing, every meta-good indexed by B_t is attached to a distinct chore in Z_t: chores allocated before round t are never reused, and all remaining meta-goods are attached in round t or later. Thus, R_t contains $|B_t|$ meta-chores and exactly $b_t = |Z_t| - |B_t|$ singleton chores.

Let $\nu = \{(i, v_i) : i \in N\}$ be a maximum-weight perfect matching in $H[N, R_t]$. By chore-maximality, every real object has strictly negative utility for every agent, whereas dummy items have utility zero. Therefore, ν uses every available dummy item; otherwise, replacing a selected real object with an unused dummy would increase its weight. By construction of Algorithm 4, R_t contains r dummy items when $t = 1$ and none afterwards. Hence, ν selects exactly q_t real objects, where q_t is defined in Equation (1).

Suppose these selected objects include p meta-chores. The other $q_t - p$ objects are singleton chores. Since R_t contains only b_t singleton chores, we have $q_t - p \leq b_t$. This is precisely the extendability condition in Equation (2). Therefore, the assignments in ν, retaining their pairings

under φ^*, form a candidate round represented by a feasible flow of value q_t in F_t. By Lemma 4.6, its profit equals the weight of ν.

Since the algorithm selects a maximum-profit flow in F_t, we have $\sum_{i\in N} u_i(\mu_i^t) \geq \sum_{i\in N} u_i(\nu_i)$. Moreover, μ^t itself is a feasible perfect matching in $H[N, R_t]$: the final pairing preserves every object assigned in round t. Thus, μ^t is a maximum-weight perfect matching in that graph. This holds for every round t. □

The next lemma ensures that EF1 holds for original items, not merely for bundled objects. Its proof uses the fact that every agent receives a real object in the final round. Indeed, $q_T = n$: if $T = 1$, then $k \geq n$ and $T = \lceil k/n \rceil$ force $k = n$ and $r = 0$; if $T > 1$, then the definition of q_t directly gives $q_T = n$.

Lemma 4.8. *If $\mu_i^T = M_j \circ c$ for some $j \in [\ell]$ and $c \in Z$, then $i \in T_j$, i.e., $u_i(M_j) \geq 0$.*

Proof. Keep each agent's underlying real chore fixed and temporarily detach the meta-goods assigned in round T. For each such meta-good M_j, choose $a_j \in \arg\max_{a\in N} u_a(M_j)$. Because M_j is a meta-good, $u_{a_j}(M_j) \geq 0$, so $a_j \in T_j$. The agents a_j are distinct because the interest sets are pairwise disjoint. Since every agent has a real chore in this round, we can attach each M_j to the chore held by a_j.

This reassignment uses exactly the same chores and meta-goods as before and changes none of the preceding rounds. It therefore remains extendable and corresponds to a feasible flow in F_T by Lemma 4.6. The total contribution from the chores is unchanged, and the contribution from each meta-good weakly increases. If some meta-good had been assigned to an agent who valued it negatively, its contribution would strictly increase. The resulting flow would then have strictly greater profit, contradicting optimality of the flow chosen in round T. □

Proposition 4.9. *Under Case I, Algorithm 4 returns an EF1 and envy-freeable allocation in polynomial time.*

Proof. Let A be the allocation returned by the algorithm and $(\mu^1, \ldots, \mu^T)$ its recorded matchings. We interpret $u_i(\mu_j^t)$ as the utility of the original items in the object, with utility zero for a dummy item.

Envy-freeability. By Lemma 4.7, every μ^t is a maximum-weight perfect matching in the remaining object collection for φ^*. For any permutation σ of N, reassigning the objects of round t according to σ is another feasible matching. It follows that

$$\sum_{i\in N} u_i(A_i) = \sum_{t=1}^{T}\sum_{i\in N} u_i(\mu_i^t) \geq \sum_{t=1}^{T}\sum_{i\in N} u_i(\mu_{\sigma(i)}^t) = \sum_{i\in N} u_i(A_{\sigma(i)}).$$

Thus A is envy-freeable by Theorem 2.6.

EF1. To prove EF1, fix any pair of agents $i, j \in N$. For every $t < T$, the object μ_j^{t+1} is still available in round t. Replacing i's matched object by this object is feasible, so maximum weight gives $u_i(\mu_i^t) \geq u_i(\mu_j^{t+1})$. Since $q_T = n$, agent i's final-round object contains an original objective chore $c_i^T \in Z$, which belongs to A_i. If that object is a singleton, then $u_i(\mu_i^T) - u_i(c_i^T) = 0$. Otherwise this difference is the value of its attached meta-good, which is non-negative by Lemma 4.8. Therefore,

$$u_i(A_i \setminus \{c_i^T\}) = \sum_{t=1}^{T-1} u_i(\mu_i^t) + u_i(\mu_i^T) - u_i(c_i^T) \geq \sum_{t=1}^{T-1} u_i(\mu_i^t) \geq \sum_{t=2}^{T} u_i(\mu_j^t) \geq \sum_{t=1}^{T} u_i(\mu_j^t) = u_i(A_j).$$

The last inequality uses $u_i(\mu_j^1) \leq 0$. Empty sums are zero, so the argument also covers $T = 1$. Removing the single original chore c_i^T eliminates all of agent i's envy, proving EF1.

Polynomial-Time Computation. There are $T = \lceil k/n \rceil$ rounds, and the flow network in each round has polynomially many nodes and arcs. In each round, negating the profits yields a minimum-cost flow problem of prescribed value $q_t \leq n$, with integral capacities. An integral optimum can be obtained by standard minimum-cost flow algorithms in time polynomial in the input bit length. Note also that the preceding item-merging procedure is polynomial-time as well. □

4.2 Case II: At Most $n-1$ Remaining Objective Chores ($|Z| \leq n-1$)

In this case, we have a collection of meta-goods $\{M_r\}_{r\in[\ell]}$ and a set of objective chores Z with $|Z| \leq n-1$. Note that when $|Z| \leq n-1$, there may not be enough chores to be combined with the meta-goods to ensure the resulting instance contains only chores. In this case, some meta-goods must be allocated on their own. This implies that, in order to allocate the meta-goods $\{M_r\}_{r\in[\ell]}$, we additionally require *good-minimality*.

Definition 4.10 (Good-minimality)**.** A non-empty subset $M' \subseteq M$ of indivisible items is said to be *good-minimal* if $u_i(M') \geq 0$ for some $i \in N$ (i.e., M' is a meta-good) and for all $j \in N$ and $g \in \{o \in M' \mid u_j(o) \geq 0\}$, $u_j(M' \setminus \{g\}) \leq 0$.

We show below the connection between good-minimality and EF1.

Lemma 4.11. *Consider any allocation $(A_i)_{i\in N}$ of items M and assume agent j receives a meta-good M' that is good-minimal. If $u_i(A_i) \geq u_i(A_j \setminus M')$ for some $i \in N$, then agent i is EF1 towards agent j.*

Proof. If $u_i(M') \leq 0$, we have $u_i(A_i) \geq u_i(A_j \setminus M') \geq u_i(A_j)$. Thus, agent i is envy-free towards agent j. Else, $u_i(M') > 0$. Since the meta-good M' is good-minimal by the statement of the lemma, there must exist a good $g \in M'$ for agent i such that $u_i(M' \setminus \{g\}) \leq 0$. As a result,

$$u_i(A_i) \geq u_i(A_j \setminus M') \geq u_i((A_j \setminus M') \cup (M' \setminus \{g\})) = u_i(A_j \setminus \{g\}),$$

meaning that agent i is EF1 towards agent j, as desired. □

We will first consider the scenario with $|Z| = 0$ in Section 4.2.1 and design an algorithm that always produces an EF1 and envy-freeable allocation, which will prove useful for the scenario with $1 \leq |Z| \leq n-1$ in Section 4.2.2.

4.2.1 Case II.1: $|Z| = 0$

In this subcase, we study the scenario that after Algorithm 2 preprocesses the given instance, there is a set of ℓ meta-goods and 0 remaining objective chores. We design Algorithm 5 to further process the output of Algorithm 2 and present its pseudocode in a more general sense so that it can be used as a subroutine for certain cases encountered later in Section 4.2.2. At a high level, taking as input a set of ℓ meta-goods $\{M_1, \ldots, M_\ell\}$ and a (possibly empty) set of subjective goods X, Algorithm 5 first obtains an updated set of meta-goods each of which also satisfies good-minimality in lines 1 to 5, and next it applies an Iterative Maximum-Weight Matching (`IMWM`) procedure to allocate the meta-goods to agents N.

Denote by $M_{\ell+1}, M_{\ell+2}, \ldots, M_{\ell+|X|}$ the meta-goods representation of the $|X|$ subjective goods from X, and update $\ell \leftarrow \ell + |X|$ accordingly. We now have a partition of items M into meta-goods

Algorithm 5: Iterative Matching for Meta-Goods

Input: A set of meta-goods $\{M_1, \ldots, M_\ell\}$ and a (possibly empty) set of subjective goods X, with $(\bigcup_{j\in[\ell]} M_j) \cup X = M$.

Output: An EF1 and envy-freeable allocation $A = (A_1, \ldots, A_n)$.

1 **foreach** $g \in X$ **do** Create the $(\ell+1)$-th meta-good $M_{\ell+1} \leftarrow \{g\}$, and update $\ell \leftarrow \ell + 1$.

2 **while** $\exists j \in [\ell], i \in N, g \in M_j$ *such that* $u_i(g) \geq 0$ *and* $u_i(M_j \setminus \{g\}) > 0$ **do**

`// Good-minimality`

3 Create the $(\ell+1)$-th meta-good $M_{\ell+1} \leftarrow \{g\}$.

4 $M_j \leftarrow M_j \setminus \{g\}$

5 $\ell \leftarrow \ell + 1$

6 $J \leftarrow \{M_1, \ldots, M_\ell\}$.

7 $(A_1, \ldots, A_n) \leftarrow$ `IMWM`$(G[N, J])$

8 **return** *allocation* A

---

9 **Function** `IMWM`$(G[N, J])$**:**

10 $A_i \leftarrow \emptyset$ for all $i \in N$.

11 $t \leftarrow 1$; $J_1 \leftarrow J$

12 **while** $J_t \neq \emptyset$ **do**

13 Compute a maximum-weight matching $\mu^t = \{(i, \mu_i^t)\}$ in $G[N, J_t]$, breaking ties in favour of larger cardinality size.

14 $A_i \leftarrow A_i \cup \mu_i^t$ for all $i \in N$.

15 $J_{t+1} \leftarrow J_t \setminus \{\mu_i^t : i \in N,\ \mu_i^t \neq \emptyset\}$

16 $t \leftarrow t + 1$

17 **return** *Allocation* A

$M_1, \ldots, M_\ell$, which may not all satisfy good-minimality. The algorithm identifies in line 2 an M_j that violates good-minimality, i.e., there exists an agent $i \in N$ and her subjective good $g \in M_j$ such that $u_i(g) \geq 0$ and $u_i(M_j \setminus \{g\}) > 0$. We then create the $(\ell+1)$-th meta-good $M_{\ell+1} \leftarrow \{g\}$, as well as update $M_j \leftarrow M_j \setminus \{g\}$ and $\ell \leftarrow \ell + 1$. Clearly, the updated M_j remains a meta-good. Since ℓ strictly increases in each iteration of the `while`-loop in lines 2 to 5 and is upper bounded by m, the `while`-loop will terminate. At its termination, all M_j's satisfy good-minimality. Let $J = \{M_1, \ldots, M_\ell\}$.

Given agents N and a set $J = \{M_1, \ldots, M_\ell\}$ of meta-goods, let $G[N, J] = (N \cup J, E)$ denote a weighted bipartite graph where there is an edge (i, M_j) for some $i \in N$ and $M_j \in J$ if $u_i(M_j) \geq 0$; the edge has weight $u_i(M_j)$.[6] For any subset $\widehat{J} \subseteq J$, let $G[N, \widehat{J}]$ denote the subgraph of $G[N, J]$ induced by N and $\widehat{J}$. In each round of `IMWM`, each agent is matched to at most one meta-good. For the first round, we set $J_1 = \{M_1, \ldots, M_\ell\}$. In each round t, we find a maximum-weight matching μ^t in $G[N, J_t]$ and agent i is matched to item μ_i^t.[7] If agent i does not match to any item, let $\mu_i^t = \emptyset$ for the ease of expression. If agent i is unmatched in round t, then $u_i(h) < 0$ for

[6] Note that $G[N, J]$ defined here is different from $H[N, J]$ defined in Section 4.1. Moreover, while we are inspired by the Iterative Matching Algorithm of Brustle et al. [2020] for goods, the way we construct $G[N, J]$ is different from theirs.

[7] Among all maximum-weight matchings, we choose one of maximum cardinality. This tie-breaking rule guarantees progress even when all available edge weights are zero and justifies the strict negativity assertion below; the EF1 and envy-freeability arguments otherwise use only maximum weight.

every meta-good $h \in J_{t+1}$. Otherwise, adding the edge (i, h) would either increase the matching weight or preserve its weight while increasing its cardinality, contradicting the selection rule. Put differently, each agent will only be allocated meta-goods of non-negative values for them. The `IMWM` procedure terminates when all items are matched to agents. Whenever J_t is non-empty, some edge exists because every remaining meta-good is valued non-negatively by at least one agent. The maximum-cardinality tie-break therefore ensures that at least one meta-good is allocated in that round, even if the maximum matching weight is zero. Hence `IMWM` terminates after at most $|J|$ rounds. We are now ready to establish the following statement.

Proposition 4.12. *Taking as input a set of meta-goods $\{M_j\}_{j\in[\ell]}$ and a (possibly empty) set of subjective goods X with $X \cup \bigcup_{j\in[\ell]} M_j = M$, Algorithm 5 always finds an EF1 and envy-freeable allocation in polynomial time.*

Proof. At the end of the first `while`-loop (lines 2 to 5) of Algorithm 5, each meta-good M_j in $J = \{M_1, \ldots, M_\ell\}$ is good-minimal. It suffices to show that the output allocation $A = (A_1, \ldots, A_n)$ of $\texttt{IMWM}(G[N, J])$ satisfies both EF1 and envy-freeability. Suppose that `IMWM` terminates in T rounds. For each round $t \in [T]$, let μ^t denote the matching in this round; moreover, each agent $i \in N$ is matched to a meta-good μ_i^t. Note that $\mu_i^t = \emptyset$ if the agent does not match to anything. At the end of `IMWM`, each agent $i \in N$ receives a bundle $A_i = \bigcup_{t\in[T]} \mu_i^t$.

Allocation A is envy-freeable. By Theorem 2.6, it suffices to show that the envy graph $\mathcal{G}_A$ has no positive-weight cycles. Take any directed cycle γ in $\mathcal{G}_A$ and assume without loss of generality that γ involves a sequence of nodes/agents $1, 2, \ldots, k$ for some $k \geq 2$. We have

$$\begin{aligned} w_A(\gamma) = \sum_{(i,j)\in\gamma} w_A(i,j) = \sum_{(i,j)\in\gamma} \big(u_i(A_j) - u_i(A_i)\big) &= \sum_{(i,j)\in\gamma} \sum_{t=1}^{T} \Big(u_i(\mu_j^t) - u_i(\mu_i^t)\Big) \\ &= \sum_{(i,j)\in\gamma} \sum_{t=1}^{T} w_{\mu^t}(i,j) = \sum_{t=1}^{T} \sum_{(i,j)\in\gamma} w_{\mu^t}(i,j) \end{aligned}$$

Let π_γ be the permutation of N under which $\pi_\gamma(i) = i + 1$ for each $i \in [k-1]$, $\pi_\gamma(k) = 1$, and $\pi_\gamma(i) = i$ for $i \in N \setminus [k]$. In each round t, μ^t is a maximum-weight matching; therefore, $\sum_{(i,j)\in\gamma} w_{\mu^t}(i,j) \leq 0$. Otherwise, consider the reassignment that gives each agent i the meta-good $\mu^t_{\pi_\gamma(i)}$, with $\emptyset$ interpreted as receiving nothing. Its total utility is strictly greater than the weight of μ^t. This reassignment may use pairs of negative utility, which are not edges of $G[N, J_t]$. Discard every such pair and leave its agent unmatched; also omit empty assignments. The remaining pairs form a feasible matching in $G[N, J_t]$, since the reassignment gives each agent at most one meta-good and assigns each meta-good at most once. Discarding negative-valued pairs cannot decrease total weight, so this matching has strictly greater weight than μ^t, a contradiction. Thus, $w_A(\gamma) \leq 0$, implying that allocation A is envy-freeable.

Allocation A is EF1. Observe that $u_i(\mu_i^t) \geq u_i(g)$ for any meta-good $g \in J_{t+1}$; otherwise, replacing the edge (i, μ_i^t) with (i, g) in μ^t results in a matching of greater weight in $G[N, J_t]$. For any $i, j \in N$,

$$\begin{aligned} u_i(A_i) = u_i(\{\mu_i^1, \ldots, \mu_i^T\}) = u_i(\mu_i^1) + \cdots + u_i(\mu_i^{T-1}) + u_i(\mu_i^T) &\geq u_i(\mu_i^1) + \cdots + u_i(\mu_i^{T-1}) \\ &\geq u_i(\mu_j^2) + \cdots + u_i(\mu_j^T). \end{aligned}$$

If $u_i(\mu_j^1) \leq 0$, clearly $u_i(A_i) \geq u_i(\mu_j^1) + u_i(\mu_j^2) + \cdots + u_i(\mu_j^T) = u_i(A_j)$, meaning that agent i is envy-free towards agent j. Else, $u_i(\mu_j^1) > 0$, meaning that $u_i(A_i) \geq u_i(A_j \setminus \mu_j^1)$. In other words,

after removing meta-good μ_j^1 from agent j's bundle, agent i would be envy-free towards j. By Lemma 4.11, agent i is EF1 towards j, as desired.

Polynomial-Time Computation. The splitting step increases the number of non-empty, pairwise disjoint meta-goods by one, so there are at most m such steps. A violating triple (j, i, g) can be found, or its absence certified, by scanning all agents and all items in the current meta-goods and testing $u_i(g) \geq 0$ and $u_i(M_j) - u_i(g) > 0$. These tests and the updates take polynomial time. After splitting, $|J| \leq m$. In each matching round, the maximum-cardinality tie-break ensures that at least one meta-good is allocated whenever any remain, so there are at most m rounds. Each round finds a maximum-weight matching, and among those one of maximum cardinality, in polynomial time on a graph with at most $n + m$ vertices. All bundle utilities and edge weights are sums of rational input utilities and have polynomial bit length. Exact arithmetic and the cardinality tie-break can therefore be implemented in polynomial time, proving the claim. □

4.2.2 Case II.2: $1 \leq |Z| \leq n - 1$

We now consider the case where Algorithm 2 outputs meta-goods $\{M_r\}_{r\in[\ell]}$ that are chore-maximal and a set of objective chores Z with $1 \leq |Z| \leq n - 1$. We design Algorithm 6 to further process those outputs and intend to find a partition of M into meta-goods $\{M_1, \ldots, M_{\widehat{\ell}}\}$, a set of subjective goods $X = \{g_1, \ldots, g_s\}$, and a set of objective chores $\widehat{Z}$ such that the following properties hold:

- Each element of $\{M_1, \ldots, M_{\widehat{\ell}}\}$ is both good-minimal and chore-maximal. Furthermore, the interest sets of the meta-goods are pairwise disjoint; that is, $T_i \cap T_j = \emptyset$ for any pair of $i, j \in \{1, \ldots, \widehat{\ell}\}$ with $i \neq j$.
- For all $i \in N$, $S \subseteq \{g_1, \ldots, g_s, M_1, \ldots, M_{\widehat{\ell}}\}$, and $c \in \widehat{Z}$, we have $u_i(\{c\} \cup S) < 0$.

To achieve the above goals, Algorithm 6 takes as input the set of meta-goods $\{M_1, \ldots, M_\ell\}$ and the set of objective chores Z outputted by Algorithm 2. Initialize $X \leftarrow \emptyset$, which will contain the subjective goods being removed from $\bigcup_{j\in[\ell]} M_j$ during the execution of Algorithm 6, $\mathcal{M} \leftarrow \{M_1, \ldots, M_\ell\}$, which is the initial set of the input meta-goods, $\widehat{Z} \leftarrow Z$, which is the initial set of the input objective chores, and an index $\widehat{\ell} \leftarrow \ell$. At a high level, the `while`-loop-condition in line 5 examines if either of the following two conditions holds:

- there exists $M_j \in \mathcal{M}$, $i \in N$ and $g \in M_j$ such that $u_i(g) \geq 0$ and $u_i(M_j \setminus \{g\}) > 0$, meaning that M_j violates good-minimality; or
- $\widehat{Z}$ is non-empty and moreover, there exists $c \in \widehat{Z}$ and $S \subseteq X \cup \mathcal{M}$ such that $u_i(S \cup \{c\}) \geq 0$.

  Note that this condition also checks if some M_j violates chore-maximality, by simply taking $S \leftarrow \{M_j\}$.

In either case, we will modify $\mathcal{M}$. We maintain that each agent values at most one $M_j \in \mathcal{M}$ non-negatively. Each iteration of the `while`-loop in lines 15 to 17 merges two meta-goods M_r and $M_{r'}$ into M_r if some agent $i' \in N$ values both M_r and $M_{r'}$ non-negatively, and removes $M_{r'}$ from $\mathcal{M}$. Clearly, the updated M_r remains a meta-good. Since $|\mathcal{M}|$ is polynomial in m and strictly decreases, each execution of the `while`-loop in lines 15 to 17 will terminate in polynomial time. At its termination, each agent values at most one $M_j \in \mathcal{M}$ non-negatively.

In the case that the `if`-statement is executed, Algorithm 6 identifies an M_j, an agent $i \in N$ and her subjective good $g \in M_j$ such that $u_i(g) \geq 0$ and $u_i(M_j \setminus \{g\}) > 0$. We then update $M_j \leftarrow$

Algorithm 6: Good-Minimal and Chore-Maximal Refinement

Input: Meta-goods $\{M_j\}_{j\in[\ell]}$ and chores Z with $1 \le |Z| \le n-1$ outputted by Algorithm 2.

Output: An updated set of meta-goods $\{M_1, \ldots, M_{\widehat{\ell}}\}$ that are both chore-maximal and good-minimal, a set of subjective goods X, and an updated set of objective chores $\widehat{Z}$.

$X \leftarrow \emptyset$
$\mathcal{M} \leftarrow \{M_1, \ldots, M_\ell\}$
$\widehat{Z} \leftarrow Z$
$\widehat{\ell} \leftarrow \ell$
while $\exists\, M_j \in \mathcal{M}, i \in N, g \in M_j$ *such that* $u_i(g) \ge 0$ *and* $u_i(M_j \setminus \{g\}) > 0$ **or** $(|\widehat{Z}| \neq 0$ **and** $\exists\, i \in N, S \subseteq X \cup \mathcal{M}, c \in \widehat{Z}$ *such that* $u_i(S \cup \{c\}) \ge 0)$ **do**
  if $\exists\, M_j \in \mathcal{M}, i \in N, g \in M_j$ *s.t.* $u_i(g) \ge 0$ *and* $u_i(M_j \setminus \{g\}) > 0$ **then** `// Good minimality.`
    $M_j \leftarrow M_j \setminus \{g\}$
    $X \leftarrow X \cup \{g\}$
  else `//` $|\widehat{Z}| \neq 0$ `and` $\exists\, i \in N, S \subseteq X \cup \mathcal{M}, c \in \widehat{Z}$ `such that` $u_i(S \cup \{c\}) \ge 0$
    $\widehat{\ell} \leftarrow \widehat{\ell} + 1$
    $M_{\widehat{\ell}} \leftarrow S \cup \{c\}$
    $\mathcal{M} \leftarrow (\mathcal{M} \setminus S) \cup \{M_{\widehat{\ell}}\}$
    $X \leftarrow X \setminus S$
    $\widehat{Z} \leftarrow \widehat{Z} \setminus \{c\}$
  while $\exists\, i' \in N, M_r, M_{r'} \in \mathcal{M}$ *with* $r \neq r'$ *such that* $u_{i'}(M_r) \ge 0$ *and* $u_{i'}(M_{r'}) \ge 0$ **do** `// Merging.`
    $M_r \leftarrow M_r \cup M_{r'}$ `//` Merge $M_{r'}$ into M_r; get an updated meta-good M_r.
    $\mathcal{M} \leftarrow \mathcal{M} \setminus \{M_{r'}\}$
$\widehat{\ell} \leftarrow |\mathcal{M}|$
Relabel all $M_j \in \mathcal{M}$ such that we have the updated set of meta-goods $\{M_1, \ldots, M_{\widehat{\ell}}\}$.
return *meta-goods* $\{M_1, \ldots, M_{\widehat{\ell}}\}$, *subjective goods* X, *and objective chores* $\widehat{Z}$

$M_j \setminus \{g\}$ and $X \leftarrow X \cup \{g\}$ by moving g from M_j to X. Clearly, the updated M_j remains a meta-good. Moreover, it can be verified easily that these operations can be done in polynomial time.

In the case that the `else`-statement is executed, Algorithm 6 identifies an agent $i \in N$, a subset $S \subseteq X \cup \mathcal{M}$ and an objective chore $c \in \widehat{Z}$ such that $u_i(S \cup \{c\}) \ge 0$. We first update $\widehat{\ell} \leftarrow \widehat{\ell} + 1$ and next create the meta-good $M_{\widehat{\ell}} \leftarrow S \cup \{c\}$. Clearly, $M_{\widehat{\ell}}$ is a meta-good. Also, we update $\mathcal{M} \leftarrow (\mathcal{M} \setminus S) \cup \{M_{\widehat{\ell}}\}$ by removing those meta-goods contained in S and adding the new meta-good $M_{\widehat{\ell}}$ to $\mathcal{M}$, as well as update $X \leftarrow X \setminus S$ and $\widehat{Z} \leftarrow \widehat{Z} \setminus \{c\}$ accordingly. We now describe how these operations can be done in polynomial time. For a fixed agent $i \in N$ and chore $c \in \widehat{Z}$, the best subset $S \subseteq X \cup \mathcal{M}$ consists of all current meta-goods that agent i values positively. Thus, the subset test is equivalent to

$$u_i(c) + \sum_{e \in X \cup \mathcal{M}} \max\{u_i(e), 0\} \ge 0;$$

when the inequality holds, those positively-valued meta-goods give a witness S directly. Clearly, this can be done in polynomial time. Furthermore, applying this idea over all agents N and all current chores in $\widehat{Z}$ runs in polynomial time.

We now prove that the outer `while`-loop (lines 5 to 17) terminates in polynomial time. In each iteration of lines 5 to 17, either the `if`- or the `else`-statement is executed. First, $|\widehat{Z}|$ strictly decreases when executing the `else`-statement and remains unchanged when lines 5 to 17 performing all other steps. The number of times the `else`-statement is executed is upper bounded by $|Z| \leq n-1$. Next, between any two consecutive executions of the `else`-statement, the cardinality size of $X \cup \bigcup_{M_j \in \mathcal{M}} M_j$, which is equal to $|M \setminus \widehat{Z}|$, stays the same. Since $|X|$ strictly increases when executing the `if`-statement, the number of times the `if`-statement is executed between any two consecutive executions of the `else`-statement is upper bounded by m. The same bound applies before the first and after the last execution of the `else`-statement, giving at most $|Z|+1$ phases and at most $m(|Z|+1)$ executions of the `if`-statement in total. Together with the fact that each component inside the `while`-loop (lines 5 to 17) runs in polynomial time, we conclude that Algorithm 6 runs in polynomial time.

At the termination of the outer `while`-loop (lines 5 to 17), if $|\widehat{Z}| = 0$, the instance can be processed by Algorithm 5 which was discussed in Section 4.2.1. Otherwise, we have $|\widehat{Z}| \geq 1$. Since the number of objective chores only decreases in our algorithm, we have $|\widehat{Z}| \leq n-1$. To summarize, after relabelling all meta-goods $M_j \in \mathcal{M}$ to obtain the updated set of meta-goods $\{M_1, \ldots, M_{\widehat{\ell}}\}$, we have the following lemma.

Lemma 4.13. *Under Case II.2, if the updated chore set $\widehat{Z}$ is non-empty, Algorithm 6 outputs, in polynomial time, a partition of the item set M into meta-goods $\{M_1, \ldots, M_{\widehat{\ell}}\}$, a set of subjective goods $X = \{g_1, \cdots, g_s\}$, and a set of objective chores $\widehat{Z} = \{c_1, \ldots, c_k\}$ with $1 \leq k \leq n-1$. These sets satisfy the following properties:*

(P1) Interest sets are pairwise disjoint, i.e., $T_i \cap T_j = \emptyset$ for all $i \neq j$.

(P2) Each element of $\{M_1, \ldots, M_{\widehat{\ell}}\}$ is both good-minimal and chore-maximal.

(P3) For every objective chore $c \in \widehat{Z}$, every agent $i \in N$, and every subset $S \subseteq \{g_1, \ldots, g_s, M_1, \ldots, M_{\widehat{\ell}}\}$, we have $u_i(\{c\} \cup S) < 0$.

Let $\mathcal{P} = \{g_1, \ldots, g_s, M_1, \ldots, M_{\widehat{\ell}}\}$ denote the singleton subjective goods and meta-goods supplied by Lemma 4.13. Treating each g_j as a singleton meta-good, every element of $\mathcal{P}$ is good-minimal. For a set $S \subseteq \mathcal{P}$, $u_i(S)$ denotes the utility of its underlying original items. The notation $c_q \circ S$, or equivalently $\{c_q\} \cup S$, denotes the bundle consisting of the original chore c_q and these items. By property (P3), every such bundle is a meta-chore.

We construct attachment sets $S_1^*, \ldots, S_k^*$ and a maximum-weight matching between the agents and the resulting meta-chores $c_1 \circ S_1^*, \ldots, c_k \circ S_k^*$. The fairness proof rests on three observations: each chore recipient values its attachments non-negatively, values every unattached meta-good negatively, and every agent values each constructed meta-chore negatively. We then allocate the remaining meta-goods by `IMWM`, as in Section 4.2.1.

Constructing the attachment sets. For each $q \in [k]$, let $S_q \subseteq \mathcal{P}$ be the set of meta-goods attached to chore c_q. Initially, $S_q = \emptyset$ for every q. The attachment sets remain pairwise disjoint, and

$$R = \mathcal{P} \setminus \bigcup_{q \in [k]} S_q$$

Algorithm 7: Greedy Expansion of Meta-Chores

Input: Agents N with additive utilities $(u_i)_{i\in N}$, objective chores $\widehat{Z} = \{c_1, \dots, c_k\}$ with $1 \le k < n$, and meta-goods $\mathcal{P}$ supplied by Lemma 4.13.

Output: Attachment sets $S_1^*, \dots, S_k^*$ and a maximum-weight matching F^* covering the constructed meta-chores.

$S_q \leftarrow \emptyset$ for every $q \in [k]$; $R \leftarrow \mathcal{P}$
while true **do**
  $B_q \leftarrow c_q \circ S_q$ for every $q \in [k]$; $\mathcal{B} \leftarrow \{B_q : q \in [k]\}$
  $w(i, B_q) \leftarrow u_i(B_q)$ for every $i \in N$ and $q \in [k]$
  foreach $i \in N$ **do**
    $R_i^+ \leftarrow \{h \in R : u_i(h) \ge 0\}$; $v_i \leftarrow \sum_{h\in R_i^+} u_i(h)$
  Compute an integral maximum-profit flow of value k in Figure 2, with secondary profit $|R_i^+|$ on each arc $s_1 \to i^-$
  $F \leftarrow \{(i, B_q) : \text{arc } i^+ \to B_q \text{ carries one unit}\}$
  Let r be the agent reached through s_1 and B_q its matched meta-chore in F
  if $R_r^+ = \emptyset$ **then**
    return $(S_1, \dots, S_k, F)$
  $S_q \leftarrow S_q \cup R_r^+$; $R \leftarrow R \setminus R_r^+$

is the set of unattached meta-goods. Write $B_q = c_q \circ S_q$ and $\mathcal{B} = \{B_q : q \in [k]\}$ for the current meta-chores. Consider the complete bipartite graph $H[N, \mathcal{B}]$ with edge weights $w(i, B_q) = u_i(B_q)$. The index of each meta-chore remains fixed as its attachment set grows.

Every matching considered in H covers all k meta-chores; it leaves $n - k$ agents unmatched. For such a matching F, let $N(F)$ be its matched agents and write

$$\mathrm{val}_w(F) = \sum_{(i,B_q)\in F} w(i, B_q), \qquad \mathrm{OPT}(w) = \max_F \mathrm{val}_w(F).$$

For each agent i, define

$$R_i^+ = \{h \in R : u_i(h) \ge 0\}, \qquad v_i = u_i(R_i^+) = \sum_{h\in R} \max\{u_i(h), 0\}.$$

Thus R_i^+ is the neighbourhood of i in the nonnegative-edge graph $G[N, R]$. At each iteration, jointly choose a matching F and one of its edges (r, B_q) to maximize

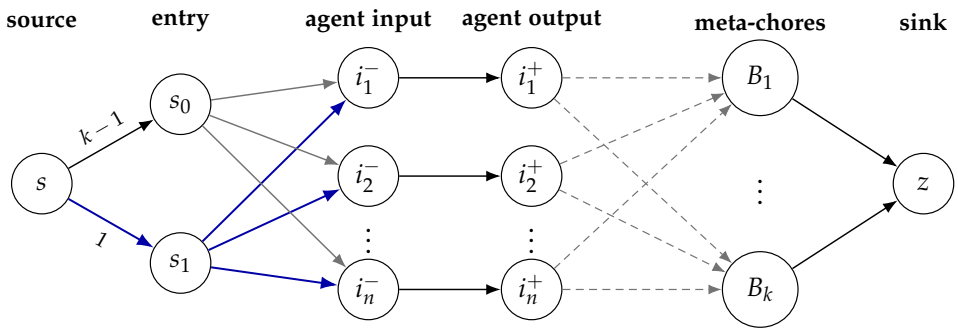
$$\mathrm{val}_w(F) + v_r, \qquad \text{breaking ties in favour of larger } |R_r^+|. \tag{3}$$

Attach R_r^+ to c_q and repeat, or terminate if $R_r^+ = \emptyset$. For later use, denote the optimal value of this choice by

$$V(w, v) = \max_{F,\, i\in N(F)} \big(\mathrm{val}_w(F) + v_i\big).$$

The choice in Equation (3) maximizes welfare after one expansion. Indeed, any set $D \subseteq R$ attached to a meta-chore assigned to i adds at most v_i to that matching's weight. Attaching R_r^+ to the selected meta-chore attains the chosen score. The attachment sets stay with their chores, while the matching is recomputed at each iteration.

After an expansion, every remaining meta-good is negative to the selected agent. Since R only shrinks, each agent can initiate at most one expansion. Hence there are at most n expansions and



Arc	Indices	Capacity	Profit
$s \to s_0$	—	$k-1$	0
$s \to s_1$	—	1	0
$s_0 \to i^-$	$i \in N$	1	0
$s_1 \to i^-$	$i \in N$	1	v_i
$i^- \to i^+$	$i \in N$	1	0
$i^+ \to B_q$	$i \in N,\ q \in [k]$	1	$u_i(B_q)$
$B_q \to z$	$q \in [k]$	1	0

Figure 2: The full network for one expansion, with $B_q = c_q \circ S_q$ for $q \in [k]$. Both entry nodes connect to every agent input, each input connects only to its own output, and every output connects to every meta-chore. Ellipses indicate the remaining agents and meta-chores. The two source-arc labels show capacities; the table specifies all capacities and profits.

one final iteration. Including zero-valued meta-goods in R_i^+ and breaking ties by $|R_i^+|$ ensures that, at termination, every chore recipient values every unattached meta-good strictly negatively. This is established as Observation 2 in the proof of Proposition 4.15 below.

Polynomial implementation. Each choice is one integral maximum-profit flow of prescribed value k in the network of Figure 2. Its vertices are the source s, the two entry nodes s_0, s_1, two nodes i^-, i^+ for each agent $i \in N$, one node B_q for each $q \in [k]$, and the sink z. The figure lists every arc family; there are no other arcs.

The prescribed flow value is k, regardless of the signs of the profits. The source capacities force exactly one unit through s_1; its agent is the selected recipient r. The k unit-capacity arcs into z force every meta-chore to be assigned, and $i^- \to i^+$ ensures distinct recipients. Conversely, any matching F and selected recipient $r \in N(F)$ define such a flow, routing r through s_1 and the other recipients through s_0. Its profit is exactly $\mathrm{val}_w(F) + v_r$. When $k = 1$, the ordinary entry simply has capacity zero. The secondary profit $|R_i^+|$ on $s_1 \to i^-$ implements the tie-break in Equation (3); any remaining ties may be broken arbitrarily. There are $O(n)$ vertices and $O(n^2)$ arcs, and at most $n+1$ flow computations. For rational utilities, clear denominators, multiply primary integer profits by $m+1$, and add the secondary profit; this implements the tie-break with polynomial bit length.

Although each expansion attaches meta-goods valued non-negatively by the selected agent, later iterations may assign the expanded meta-chore to another agent. The following lemma ensures that its final recipient still values the attachment set non-negatively.

Lemma 4.14 (Non-negative attachment utility)**.** *At every stage of Algorithm 7, every maximum-weight matching assigns each meta-chore $B_q = c_q \circ S_q$ to an agent i with $u_i(S_q) \geq 0$.*

The proof in Appendix A uses an alternating-path bound to maintain a stronger invariant: resetting any one expanded meta-chore to its original chore, even followed by the best next expansion, cannot improve on the current maximum matching weight. We apply this lemma in Observation 1 below.

The resulting allocation. Let $(S_1^*, \ldots, S_k^*, F^*)$ be the output of Algorithm 7. Define

$$N^d = N \setminus N(F^*), \qquad R^* = \mathcal{P} \setminus \bigcup_{q \in [k]} S_q^*.$$

Relabel the agents in $N \setminus N^d$ so that agent i is matched to $c_i \circ S_i^*$, for each $i \in [k]$, and label the remaining agents so that $N^d = \{k+1, \ldots, n\}$. Allocate $A_i = \{c_i\} \cup S_i^*$ to each $i \in [k]$. Allocate the remaining meta-goods R^* to the agents in N^d using `IMWM` in $G[N^d, R^*]$, as in Case II.1. In each round, choose a maximum-cardinality matching among maximum-weight matchings.

Introduce $n - k$ dummy chores $d_1, \ldots, d_{n-k}$, each valued at zero by every agent, and define

$$\widetilde{J}(S_1^*, \ldots, S_k^*) = \{d_1, \ldots, d_{n-k}\} \cup \{c_i \circ S_i^* : i \in [k]\}.$$

Assigning these dummy chores to the agents in N^d extends F^* to a perfect matching in $H[N, \widetilde{J}(S_1^*, \ldots, S_k^*)]$.

Proposition 4.15. *Under Case II.2, the allocation $A = (A_1, \ldots, A_n)$ described above is complete, EF1, and envy-freeable. The allocation can be computed in polynomial time.*

Proof. We first establish three observations used in the fairness proof.

Observation 1: Each agent $i \in N \setminus N^d$ derives non-negative total utility from its attachment set, i.e., $u_i(S_i^*) \geq 0$. Lemma 4.14 shows that every maximum-weight matching assigns each meta-chore to an agent who values its attachment set non-negatively. Applying it to F^* proves the observation. This concerns the total utility of S_i^*; individual meta-goods in that set may have negative utility for i.

Observation 2: Each agent $i \in N \setminus N^d$ derives negative utility from every remaining meta-good in $\mathcal{P} \setminus \bigcup_{q \in [k]} S_q^*$. Suppose that $R_i^+ \neq \emptyset$ for some matched agent i at termination. If $v_i^* > 0$, selecting i with the same matching F^* would increase the expansion score. If $v_i^* = 0$, it would leave the score unchanged and improve the tie-break, since the selected agent r has $|R_r^+| = 0$. Both cases contradict the algorithm's choice. Hence $R_i^+ = \emptyset$ for every chore recipient i.

Observation 3: For each agent $r \in N$ and any $i \in [k]$, we have $u_r(A_i) = u_r(\{c_i\} \cup S_i^*) < 0$. This follows from property (P3) of Lemma 4.13, since $c_i \in \widehat{Z}$ and $S_i^* \subseteq \mathcal{P}$.

Each remaining meta-good has a non-negative valuer in N. By Observation 2, this valuer belongs to N^d. Moreover, good-minimality is preserved when restricting the agents to N^d. Thus Algorithm 5 applies to $G[N^d, R^*]$ without further splitting. Since $\mathcal{P}$ and $\widehat{Z}$ partition M, and the attachment sets are pairwise disjoint, the allocation A is complete.

Allocation A is EF1. Fix an agent $i \in N \setminus N^d$. By Observation 1, removing the single original chore c_i leaves non-negative utility. Every other chore bundle has negative utility for i by Observation 3. Each bundle allocated to an agent in N^d consists of remaining meta-goods, so it has non-positive utility for i by Observation 2. Consequently, for every $j \in N$,

$$u_i(A_i \setminus \{c_i\}) = u_i(S_i^*) \geq 0 \geq u_i(A_j).$$

Hence every agent in $N \setminus N^d$ satisfies EF1.

The agents in N^d satisfy EF1 among themselves by Proposition 4.12. Each receives non-negative utility, whereas every chore bundle has negative utility for that agent by Observation 3. Thus no agent in N^d envies a chore recipient, proving that A is EF1.

The output matching is maximum-weight. We now establish the maximum-weight matching property. Let w^* and v^* be the weights and residual utilities at termination. The selected agent r has $R_r^+ = \emptyset$, and hence $v_r^* = 0$. Since F^* and r attain the best expansion score,

$$V(w^*, v^*) = \mathrm{val}_{w^*}(F^*) \leq \mathrm{OPT}(w^*) \leq V(w^*, v^*).$$

The last inequality holds because all residual utilities are non-negative. Thus $\mathrm{val}_{w^*}(F^*) = \mathrm{OPT}(w^*)$: the output matching maximizes total utility for the fixed constructed meta-chores. Since the dummy chores have zero utility, its extension is a maximum-weight perfect matching in $H[N, \widetilde{J}(S_1^*, \ldots, S_k^*)]$.

Allocation A is envy-freeable. Decompose A into

$$Q = (A_1, \ldots, A_k, \emptyset, \ldots, \emptyset), \qquad W = (\emptyset, \ldots, \emptyset, A_{k+1}, \ldots, A_n).$$

The maximum-weight perfect matching property proved above implies that Q maximizes welfare among permutations of its bundles. By Observation 2, the agents in $N \setminus N^d$ are isolated in $G[N, R^*]$ and in every subsequent residual graph. Hence the same sequence of maximum-weight matchings used on N^d can also be used on all of N, giving allocation W. By Proposition 4.12 and Theorem 2.6, W also maximizes welfare among permutations of its bundles. For every permutation σ of N, additivity therefore gives

$$\sum_{i \in N} u_i(A_i) = \sum_{i \in N} u_i(Q_i) + \sum_{i \in N} u_i(W_i) \geq \sum_{i \in N} u_i(Q_{\sigma(i)}) + \sum_{i \in N} u_i(W_{\sigma(i)}) = \sum_{i \in N} u_i(A_{\sigma(i)}).$$

By Theorem 2.6, A is envy-freeable.

Polynomial-time computation. The refinement is polynomial by Lemma 4.13, and Algorithm 7 uses at most $n + 1$ flow computations of polynomial size. In the final `IMWM`, every remaining meta-good has an incident edge in the nonnegative-edge graph. The maximum-cardinality tie-break therefore allocates at least one meta-good in every nonempty round, giving at most $|\mathcal{P}| \leq m$ rounds. All weights are sums of input utilities, and exact comparisons and tie-breaking have polynomial bit complexity. □

Combining Propositions 4.9, 4.12 and 4.15, we establish Theorem 4.2.

5 Discussion

In this paper, we have studied the general fair division problem that involves indivisible goods and chores as well as a cake, and show the existence of EFM allocations. To establish this result, we prove that with indivisible goods and chores, there always exists an allocation that is both EF1 and envy-freeable. Moreover, we devise polynomial-time algorithms to compute such an allocation. Our findings uncover an interesting connection between solution concepts of envy-freeness, envy-freeability, EF1, and EFM that have been investigated in closely related but separately developed lines of research.

A potential direction for future work is to consider the more general class of valuations beyond additive. For instance, with only indivisible items, Bhaskar et al. [2021] devised an

algorithm that always finds an EF1 allocation for doubly monotone instances (cf. Footnote 2). Our iterative matching techniques can no longer extend beyond additive valuations. This raises the following intriguing questions. Does an EFM allocation always exist for agents with doubly monotone valuations over indivisible items?

Acknowledgements

This work was partially supported by the NSF-CSIRO grant on "Fair Sequential Collective Decision Making", by the ARC Laureate Project FL200100204 on "Trustworthy AI", and by JST ERATO Grant Number JPMJER2301.

We would like to thank Shengxin Liu and the anonymous reviewers for their valuable feedback.

Generative AI tools, especially ChatGPT (GPT-6 Astra, accessed October 2026), assisted in writing the polynomial-time arguments.

References

Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987.

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

Enriqueta Aragones. A derivation of the money Rawlsian solution. *Social Choice and Welfare*, 12 (3):267–276, 1995.

Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 416–427, 2016a.

Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 454–464, 2016b.

Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation. *Operations Research Letters*, 48(5):573–578, 2020.

Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi-Agent Systems*, 36(1):3:1–3:21, 2022.

Haris Aziz, Zixu He, Xinhang Lu, and Kaiyang Zhou. Fair allocation of divisible goods under non-linear valuations. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 170–178, 2025.

Haris Aziz, Xiaolin Bu, Xinhang Lu, Simon Mackenzie, Mashbat Suzuki, Biaoshuai Tao, and Toby Walsh. Best-of-both-worlds fairness for mixed goods and chores. In *Proceedings of the 38th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2027. Forthcoming.

Siddharth Barman and Paritosh Verma. Introspectively envy-free and efficient allocation of indivisible mixed manna. *CoRR*, abs/2509.18673, 2026.

Siddharth Barman, Vishwa Prakash HV, Aditi Sethia, and Mashbat Suzuki. Fair and efficient allocation of indivisible mixed manna. In *Proceedings of the 21th Conference on Web and Internet Economics (WINE)*, pages 467–483, 2025.

Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. Fair division of mixed divisible and indivisible goods. *Artificial Intelligence*, 293:103436, 2021a.

Xiaohui Bei, Shengxin Liu, Xinhang Lu, and Hongao Wang. Maximin fairness with mixed divisible and indivisible goods. *Autonomous Agents and Multi-Agent Systems*, 35(2):34:1–34:21, 2021b.

Xiaohui Bei, Shengxin Liu, and Xinhang Lu. Fair division with subjective divisibility. *Games and Economic Behavior*, 151:127–147, 2025.

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.

Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. On approximate envy-freeness for indivisible chores and mixed resources. In *Proceedings of the 24th International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 1:1–1:23, 2021.

Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaya. Competitive division of a mixed manna. *Econometrica*, 85(6):1847–1871, 2017.

Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

Johannes Brustle, Jack Dippel, Vishnu V. Narayan, Mashbat Suzuki, and Adrian Vetta. One dollar each eliminates envy. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 23–39, 2020.

Xiaolin Bu, Zihao Li, Shengxin Liu, Xinhang Lu, and Biaoshuai Tao. Best-of-both-worlds fair allocation of indivisible and mixed goods. In *Proceedings of the 20th Conference on Web and Internet Economics (WINE)*, pages 277–294, 2024.

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

Ioannis Caragiannis and Stavros Ioannidis. Computing envy-freeable allocations with limited subsidies. In *Proceedings of the 17th International Conference on Web and Internet Economics (WINE)*, pages 522–539, 2021.

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):12:1–12:32, 2019.

Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. A complementary pivot algorithm for competitive allocation of a mixed manna. *Mathematics of Operations Research*, 48(3):1630–1656, 2023.

Ya'akov (Kobi) Gal, Moshe Mash, Ariel D. Procaccia, and Yair Zick. Which is the fairest (rent division) of them all? *Journal of the ACM (JACM)*, 64(6):39:1–39:22, 2017.

Hiromichi Goko, Ayumi Igarashi, Yasushi Kawase, Kazuhisa Makino, Hanna Sumita, Akihisa Tamura, Yu Yokoi, and Makoto Yokoo. A fair and truthful mechanism with limited subsidy. *Games and Economic Behavior*, 144:49–70, 2024.

Hao Guo, Weidong Li, and Bin Deng. A survey on fair allocation of chores. *Mathematics*, 11(16): 3616, 2023.

Claus-Jochen Haake, Matthias G. Raith, and Francis Edward Su. Bidding for envy-freeness: A procedural approach to n-player fair-division problems. *Social Choice and Welfare*, 19(4):723—749, 2002.

Daniel Halpern and Nisarg Shah. Fair division with subsidy. In *Proceedings of the 12th International Symposium on Algorithmic Game Theory (SAGT)*, pages 374–389, 2019.

Wentao He and Biaoshuai Tao. EFX for additive chores: Nonexistence, Pareto incompatibility, and bi-valued existence. In *Proceedings of the 38th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2027. Forthcoming.

Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Fairly dividing mixtures of goods and chores under lexicographic preferences. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 152–160, 2023.

Yasushi Kawase, Kazuhisa Makino, Hanna Sumita, Akihisa Tamura, and Makoto Yokoo. Towards optimal subsidy bounds for envy-freeable allocations. *Artificial Intelligence*, 348:104406, 2025.

Flip Klijn. An algorithm for envy-free allocations in an economy with indivisible objects and money. *Social Choice and Welfare*, 17(2):201–215, 2000.

Rucha Kulkarni, Ruta Mehta, and Setareh Taki. On the PTAS for maximin shares in an indivisible mixed manna. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5523–5530, 2021a.

Rucha Kulkarni, Ruta Mehta, and Setareh Taki. Indivisible mixed manna: On the computability of MMS + PO allocations. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 683–684, 2021b.

Bo Li, Zihao Li, Shengxin Liu, and Zekai Wu. Allocating mixed goods with customized fairness and indivisibility ratio. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2868–2876, 2024a.

Zihao Li, Shengxin Liu, Xinhang Lu, and Biaoshuai Tao. Truthful fair mechanisms for allocating mixed divisible and indivisible goods. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2808–2816, 2023.

Zihao Li, Shengxin Liu, Xinhang Lu, Biaoshuai Tao, and Yichen Tao. A complete landscape for the price of envy-freeness. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1183–1191, 2024b.

Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

Shengxin Liu, Xinhang Lu, Mashbat Suzuki, and Toby Walsh. Mixed fair division: A survey. *Journal of Artificial Intelligence Research*, 80:1373–1406, 2024.

Xinhang Lu, Simon Mackenzie, and Mashbat Suzuki. Optimal subsidy bounds for goods and chores: One dollar each suffices. *CoRR*, abs/2607.10089, 2026.

Shaily Mishra, Manisha Padala, and Sujit Gujar. Fair allocation of goods and chores – tutorial and survey of recent results. *CoRR*, abs/2307.10985, 2023.

Trung Thanh Nguyen and Jörg Rothe. Complexity results and exact algorithms for fair division of indivisible items: A survey. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6732–6740, 2023.

Koichi Nishimura and Hanna Sumita. Envy-freeness and maximum Nash welfare for mixed divisible and indivisible goods. *Mathematical Social Sciences*, 138:102449, 2025.

Ariel D. Procaccia. Cake cutting: Not just child's play. *Communications of the ACM*, 56(7):78–87, 2013.

Jack Robertson and William Webb. *Cake-Cutting Algorithm: Be Fair If You Can*. A K Peters/CRC Press, 1998.

Hugo Steinhaus. Sur la division pragmatique. *Econometrica*, 17:315–319, 1949.

Walter Stromquist and Douglas R Woodall. Sets on which several measures agree. *Journal of Mathematical Analysis and Applications*, 108(1):241–248, 1985.

Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *The American Mathematical Monthly*, 106(10):930–942, 1999.

Warut Suksompong. Constraints in fair division. *ACM SIGecom Exchanges*, 19(2):46–61, 2021.

Warut Suksompong. Weighted fair division of indivisible items: A review. *Information Processing Letters*, 187:106519, 2025.

Lars-Gunnar Svensson. Large indivisibles: An analysis with respect to price equilibrium and fairness. *Econometrica*, 51(4):939–954, 1983.

A Attachment Values Under Greedy Expansion

We prove Lemma 4.14 using an alternating-path bound and a reset invariant. Throughout this appendix, a matching covers all k real meta-chores in $H[N, \mathcal{B}]$ and leaves $n - k$ agents unmatched; dummy chores are not included. The quantities $\mathrm{val}_w(F)$, $\mathrm{OPT}(w)$, and $V(w, v)$ are as defined in Section 4.2.2.

Lemma A.1 (One alternating-path bound)**.** *Suppose $\widehat{w}$ differs from w only on edges incident to one meta-chore. For any non-negative vector v, any matching F, and distinct agents $i, j \in N(F)$,*

$$\mathrm{val}_{\widehat{w}}(F) + v_i + v_j \leq V(\widehat{w}, v) + V(w, v) - \mathrm{OPT}(w). \tag{4}$$

Proof. Fix a maximum-weight matching F^0 for w. Selecting any $a \in N(F^0)$ gives $\mathrm{OPT}(w) + v_a \le V(w, v)$. Thus, if either i or j is matched in F^0, selecting the other agent in $V(\widehat{w}, v)$ proves the bound.

Otherwise, i and j are the entering endpoints of two distinct alternating paths in $F \triangle F^0$: every meta-chore is matched in both matchings, so each path introduces just one new agent. At most one of these paths contains the meta-chore whose weights changed. Choose the other path, say the one entering j, and let g be the increase in w-weight when it is applied to F^0. The resulting matching contains j, so $\mathrm{OPT}(w) + g + v_j \le V(w, v)$. Reverting this path in F leaves a matching that still contains i and has $\widehat{w}$-weight $\mathrm{val}_{\widehat{w}}(F) - g$. Hence $\mathrm{val}_{\widehat{w}}(F) - g + v_i \le V(\widehat{w}, v)$. Adding these inequalities proves (4). □

For each $q \in [k]$, let w^{-q} be obtained from w by replacing $w(i, B_q)$ by $u_i(c_q)$ for every $i \in N$. This resets just the meta-chore B_q to its original chore c_q. Its attachments are not returned to R, so the vector v stays fixed. We maintain, separately for every attachment set $S_q \neq \emptyset$, the invariant

$$V(w^{-q}, v) \le \mathrm{OPT}(w). \tag{5}$$

Thus, after resetting any one previously expanded meta-chore, even the best next expansion cannot exceed the current maximum welfare.

Proof of Lemma 4.14. We first prove (5) by induction on the number of expansions. It holds initially because every attachment set is empty. Suppose $D = R_r^+$ is attached to c_p, and use primes for the new state. The choice in Algorithm 7 gives

$$\mathrm{OPT}(w') = V(w, v). \tag{6}$$

Indeed, the selected matching attains the right-hand side after the expansion. Conversely, in any matching of the new meta-chores, if agent i receives B_p, the weight is the old matching weight plus $u_i(D) \le v_i$, and hence is at most $V(w, v)$. Also, $v'_i \le v_i$ and

$$u_i(D) \le v_i - v'_i \le v_i \qquad (i \in N), \tag{7}$$

since $v_i - v'_i = \sum_{h \in D} \max\{u_i(h), 0\}$.

Consider a previously expanded meta-chore B_q with $q \neq p$. Take any matching F, let i receive B_p, and select any $j \in N(F)$. Its new score after resetting B_q is

$$\mathrm{val}_{(w')^{-q}}(F) + v'_j = \mathrm{val}_{w^{-q}}(F) + u_i(D) + v'_j.$$

If $i = j$, (7) bounds this by $\mathrm{val}_{w^{-q}}(F) + v_i \le V(w^{-q}, v) \le \mathrm{OPT}(w) \le \mathrm{OPT}(w')$. If $i \neq j$, Lemma A.1, with $\widehat{w} = w^{-q}$, bounds it by

$$\mathrm{val}_{w^{-q}}(F) + v_i + v_j \le V(w^{-q}, v) + V(w, v) - \mathrm{OPT}(w) \le V(w, v) = \mathrm{OPT}(w').$$

Maximizing over F and j proves the new invariant for $q \neq p$.

For $q = p$, resetting removes the new attachments as well. If B_p was already expanded, $(w')^{-p} = w^{-p}$ and $v' \le v$ give

$$V((w')^{-p}, v') \le V(w^{-p}, v) \le \mathrm{OPT}(w) \le \mathrm{OPT}(w').$$

If this is its first expansion, $(w')^{-p} = w$, so $V((w')^{-p}, v') \le V(w, v) = \mathrm{OPT}(w')$. This completes the induction.

Finally, take any maximum-weight matching F and let i receive $B_q = c_q \circ S_q$. If $S_q = \emptyset$, the claim is immediate. Otherwise, resetting B_q reduces the weight of this same matching by exactly $u_i(S_q)$. Selecting i in the reset problem and applying (5) gives

$$\mathrm{OPT}(w) - u_i(S_q) + v_i = \mathrm{val}_{w^{-q}}(F) + v_i \leq V(w^{-q}, v) \leq \mathrm{OPT}(w).$$

Therefore $u_i(S_q) \geq v_i \geq 0$. In particular, this holds for every chore recipient in the final matching F^*, proving Observation 1 in the proof of Proposition 4.15. □