Approximately Envy-free and Equitable Allocations of Indivisible Items for Non-monotone Valuations*

Vittorio Bilò¹, Martin Loebl², and Cosimo Vinci¹

¹Department of Mathematics and Physics "Ennio De Giorgi", University of Salento, Italy, {vittorio.bilo,cosimo.vinci}@unisalento.it

²Department of Applied Mathematics, "Charles" University of Prague, Czech Republic, loebl@kam.mff.cuni.cz

Abstract

We revisit the setting of fair allocation of indivisible items among agents with heterogeneous, non-monotone valuations. We explore the existence and efficient computation of allocations that approximately satisfy either envy-freeness or equity constraints. Approximate envyfreeness ensures that each agent values her bundle at least as much as those given to the others, after some (or any) item removal, while approximate equity guarantees roughly equal valuations among agents, under similar adjustments. As a key technical contribution of this work, by leveraging fixed-point theorems (such as Sperner's Lemma and its variants), we establish the existence of envy-free-up-to-one-good-and-one-chore (EF1_a) and equitable-up-to-one-good-and-one-chore (EQ1_a) allocations, for non-monotone valuations that are always either non-negative or non-positive. These notions represent slight relaxations of the well-studied envy-free-up-to-oneitem (EF1) and equitable-up-to-one-item (EQ1) guarantees, respectively. Our existential results hold even when items are arranged in a path and bundles must form connected sub-paths. The case of non-positive valuations, in particular, has been solved by proving a novel multicoloring variant of Sperner's Lemma that constitutes a combinatorial result of independent interest. In addition, we also design a polynomial-time dynamic programming algorithm that computes an $\mathrm{EQ1}_g^c$ allocation. For monotone non-increasing valuations and path-connected bundles, all the above results can be extended to EF1 and EQ1 guarantees as well. Finally, we provide existential and computational results for certain stronger up-to-any-item equity notions under objective valuations, where items are partitioned into goods and chores.

1 Introduction

Fair division Steinhaus [1948], the field that studies how to fairly allocate resources among a set of agents, has numerous applications across a variety of real-life scenarios, such as divorce settlements, credit assignment, and rent and land division, to name a few. Although fair division has been studied for decades in mathematics and economics, the field has attracted increasing attention from the computer science and AI community in recent years, driven by the flourishing of new fairness concepts and the demand for computationally efficient solutions overcoming the inherent impossibility of achieving optimal fairness guarantees.

Two prominently investigated notions of fairness are *envy-freeness* Foley [1966] and *equitability* Dubins and Spanier [1961]. An allocation (of items to agents) is *envy-free* (EF) if the value that every agent gives to her assigned

^{*}This work was partially supported by: the Horizon EU Framework Programme under Grant agreement No 101183743 (AGATE); the PNRR MIUR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI; the MUR - PNRR IF Agro@intesa; the Project SERICS (PE00000014) under the NRRP MUR program funded by the EU – NGEU; GNCS-INdAM.

bundle (of items) is not less than the value she gives to the bundle assigned to any other agent; it is *equitable* (EQ) if the value that every agent gives to her assigned bundle is not less than the value that the other agents assign to their respective bundles. So, the two notions coincide when agents have identical valuations.

The nature of valuation functions tremendously impacts the solution of a fair division problem. When an agent's valuation is monotone non-decreasing (resp., non-increasing), items are said to be *goods* (resp., *chores*) for the agent; when it is non-monotone, items are said to be *mixed*. Notable special cases of non-monotone valuations include *non-negative* (resp. *non-positive*) valuations, where every bundle yields a non-negative (resp. non-positive) value, and *objective* valuations, in which items can be partitioned into goods and chores. Valuations, either monotone and non-monotone, are *additive*, when each item has a value and the value of a bundle is defined by the sum of the values of its items.

While objective valuations have been widely studied Aziz et al. [2022], Barman et al. [2024], less work has been done for non-negative or non-positive ones, despite their potential applicability in numerous settings. These valuations, for instance, arise when items correspond to nodes in an edge-weighted graph with exclusively positive or negative weights, allocations are interepreted as clusterings, and the value of each bundle/cluster is then determined by factors such as the total weight of internal or cut edges, or other graph-connectivity properties. This class of settings has interesting connections with clustering problems where further fairness guarantees are required (see, e.g., Dinitz et al. [2022], Chierichetti et al. [2017], Schwartz and Zats [2022]).

Either envy-free or equitable allocations are guaranteed to exist under non-negative or non-positive valuations in the setting of *divisible items*, where items can be arbitrarily split among subsets of agents Dubins and Spanier [1961], Stromquist [1980], Woodall [1980], Su [1999], Cechlárová et al. [2013], Chèze [2017], Aumann and Dombb [2015], Bhaskar et al. [2025]. In contrast, in presence of *indivisible items* which have to be integrally assigned to any of the agents, existence cannot be guaranteed even for two agents with additive monotone valuations. To overcome this limitation, a number of relaxations have been proposed in the literature. They allow the removal of one item from a bundle when

agents perform bundle comparisons. The removal strategy clearly depends on the nature of the considered items. When an agent compares her bundle A against another bundle B, she can choose between removing a chore from A or removing a good from B. These relaxations have given rise to the notions of *envy-freeness-up-to-any-good* (EFX) Caragiannis et al. [2019], *envy-freeness-up-to-one-good* (EF1) Lipton et al. [2004], Budish [2011], *equitability-up-to-any-good* (EQX) Gourvès et al. [2014] and *equitability-up-to-one-good*, one means that the fairness property holds irrespectively of which item is selected for removal; by *up-to-one-good*, instead, the property must hold for at least one removed item. Clearly, fairness *up-to-any-good* implies fairness *up-to-one-good*.

An interesting and largely studied generalization of fair division assumes the existence of an *item graph* modeling proximity relationships among items. Every bundle has to induce a connected subgraph and an item removal is allowed only if it does not disconnect the induced subgraph Bouveret et al. [2017], Bilò et al. [2022], Igarashi [2023], Misra et al. [2021], Suksompong [2019]. With this respect, the *path constraint* assumes that the item graph is a connected path.

Our Contribution.

Given the lack of positive results for non-monotone valuations under standard approximate fairness notions, often due to impossibility barriers (see, e.g., Amanatidis et al. [2023], Barman et al. [2024]), we study slight relaxations of EF1, EQ1 and EQX, denoted by EF1^c_q (envy-free-upto-one-good-and-one-chore), $\mathrm{EQ1}_g^c$ (equitable-up-to-one-good-and-one-chore), and EQX_g^c (equitable-up-to-any-good-or-any-chore), respectively. While EF1 and EQ1 require the envy-freeness and equitability properties, respectively, to hold upon the removal of at most one item from "some" bundle, $\mathrm{EF1}_q^c$ and $\mathrm{EQ1}_q^c$ allow the properties to hold when removing at most one item from "each" bundle (i.e., for each agent, at most one chore from her own bundle and at most one good from others' bundles). Similarly, while EQX requires the equitability property to hold regardless of which item is removed (i.e., whether it is a good or a chore), EQX_q^c allows it to hold for the removal of either goods only or chores only (see Section 2). We obtain positive results on the existence and computation

of allocations satisfying the above fairness criteria across several broad classes of non-monotone valuations.

Results for non-negative or non-positive valuations: Our main contribution concerns the existence and computation of allocations which are fair up-to-one-good-andone-chore, under non-negative or non-positive valuations. In particular, we show that an EQ1 $_a^c$ allocation always exists and can be computed in polynomial time, for both non-negative (Theorem 3.3 and 3.4) and non-positive valuations (Theorem 4.2). For $EF1_a^c$ allocations, we only show existence (Theorem 3.5 and Theorem 4.3). The existence and computation of approximately envy-free or equitable allocations under non-monotone valuations is one of the major open problems in fair division (see, e.g., the surveys by Amanatidis et al. [2023], Liu et al. [2024]). Our results represent a significant step forward in this direction, due to the generality of the non-monotone valuations we consider (either non-negative or non-positive) and the fairness guarantees achieved (requiring the removal of at most one good and one chore).

It is worth noting that our results continue to hold even under path constraints, and in this sense, they generalize the findings of Bilò et al. [2022], Igarashi [2023], Misra et al. [2021], Suksompong [2019], which apply only to monotone non-decreasing valuations, and the results of Bouveret et al. [2017, 2019], which address the computational problem of finding EF and EQ allocations (that, in general, may not exist). Our existential results are obtained using Sperner's Lemma Sperner [1928] or its variants, and represent a non-trivial generalization of the approaches previously explored in Bilò et al. [2022], Igarashi [2023], as handling the non-monotonicity of the valuations poses significant technical challenges (such as the derivation and analysis of the cases described in Figure 3 of Appendix G). In particular, to address the specific case of non-positive valuations, we introduce and exploit a novel multi-coloring variant of Sperner's Lemma (Theorem 4.1), that constitutes a combinatorial result of independent interest. The computational results have been obtained by means of the dynamic-programming paradigm. Finally, when valuations are monotone (non-decreasing or non-increasing), our results extend to the stronger EQ1 and EF1 guarantees under path constraints, thereby generalizing the results of Igarashi [2023], Bilò et al. [2022], Misra et al. [2021], which exibith EF1 and EQ1 allocations under monotone non-decreasing valuations and path constraints.

Results for objective valuations: To complete the picture for non-monotone valuations, we also consider fair allocations under objective valuations, and in most of the cases the obtained results hold even under the up-toany-good-or-any-chore approximation guarantee. In particular, we show that an EQX_q^c allocation always exists and can be computed in pseudo-polynomial time (Theorem E.1 of Appendix E), via a simple variant of the localsearch approach adopted by Barman et al. [2024]. This result extends to EQX when valuations are monotone nonincreasing (Corollary E.1 of Appendix E), thereby generalizing the result of Barman et al. [2024], which holds only for monotone non-decreasing valuations. For valuations that are both objective and additive, we strengthen the above computational result by showing that an EQX $_a^c$ allocation can be found through a simple and more efficient greedy algorithm (Theorem E.3 of Appendix E). This result strengthens the findings of Hosseini and Sethia [2025], who establish existence and polynomialtime computability of EQ1 allocations under additive objective valuations. With this respect, we also show that a slight generalization of the polynomial-time algorithm proposed by Hosseini and Sethia [2025], continues to produce EO1 allocations even for objective valuations that are non-additive (Theorem E.2 of Appendix E). It is worth noting that, just as EQX_q^c is considered a relaxation of EQX, similar relaxations for EFX have been studied for objective Hosseini et al. [2023] or identical non-monotone valuations Bérczi et al. [2024]. However, these works only show non-existence of these relaxed notions.

Most of the technical details on non-negative and non-positive valuations are deffered to Appendix B-D, while those regarding objective valuations are all deferred to Appendix E-F. Table 1 summarizes both our results and related work on the considered fairness notions and classes of valuations, also specifying the cases that remain unresolved and are therefore posed as open problems.

Further Related Work. Lipton et al. [2004] show that an EF1 allocation always exists and can be efficiently computed. Their result has been extended to objective valuations by Aziz et al. [2022], Bhaskar et al. [2021], Bérczi et al. [2024]. Again under obejctive valuations, Hosseini and Sethia [2025] show existence and polynomial time

	Gen	NNeg	NPos	NDec	NInc	Obj
$\frac{\mathbf{EQ1}_g^c}{\mathbf{EF1}_g^c}$	xa	✓ P,cns	✓ P,cns	√P,cns	✓ P,cns	√P
$\overline{\mathrm{EF1}_g^c}$	$\frac{x_a}{?_a}$	√cns	√cns	√cns	√cns	$\overline{\checkmark^{\mathrm{P}}}$
EQ1	xa	?	?	√P,cns	√P,cns	√P
EF1	$\overline{?_a}$?	?	√ ^{cns}	√cns	$\overline{\checkmark^{P}}$
\mathbf{EQX}_g^c	xa	?	?	√P-	✓P-	$\checkmark^{P-}, \checkmark^{P}_{ao}$
EQX	xa	?	?	√P-	√P-	<u>xa</u>
EFX	$\frac{x_a}{x_a}$	$\overline{?_{\mathrm{a}}}$	$\overline{?_a}$?a	?a	$\overline{x_a}$

Table 1: Landscape of results for the considered fairness notions. Gen, NNeg, NPos, NDec, NInc, and Obj stand, respectively, for General, Non-negative, Non-positive, Non-decreasing, Non-increasing, and Objective valuations. Gray-highlighted results refer to our findings. \checkmark , x and ? mean respectively "it always exists", "it does not generally exist" and "existence is an open problem". Subscript "a" (resp., "ao") means that the the result of type x or ? (resp., \checkmark) holds even (resp., only) for additive valuations. Superscript "P" (resp., "P-") means that the existence can be obtained via a polynomial (resp., pseudo-polynomial) algorithm, and superscript "cns" means that the result holds even under path constraints.

computation of an EQ1 allocation; conversely, they show that, under additive non-objective valuations, EQ1 allocations may not exist. Existence and computation of Pareto optimal EF1 or EQ1 allocations have been studied by Caragiannis et al. [2019], Freeman et al. [2019, 2020], Garg and Murhekar [2024]. EF1 and EQ1 allocations under non-objective valuations have been determined for restricted cases only, and their general existence and computation is a major open problem (see surveys by Amanatidis et al. [2023], Liu et al. [2024]).

EQX allocations were first proved to exist for additive monotone non-decreasing valuations in Gourvès et al. [2014]. Efficient algorithms computing one have been later designed by Freeman et al. [2019, 2020], also covering the non-increasing case. Existence and pseudopolynomial time computation of an EQX allocation for monotone non-decreasing (and possibly non-additive) valuations has been shown in Barman et al. [2024]. This result is complemented by proving that, if one drops the monotonicity assumption, EQX allocations may not exist even for two agents with additive valuations. So, our results state that, slightly relaxing EQX to EQX $_g^c$ suffices to recover existence under non-monotone valuations, as long as they are objective (see Table 1).

The EFX criterion was introduced by Caragiannis et al. [2019], and its existence, under additive non-negative or non-positive valuations, has been addressed only in spe-

cific cases and it remains a major open problem in fair division (see surveys by Amanatidis et al. [2023], Liu et al. [2024]); instead, for objective additive valuations, an EFX allocation might not exist Hosseini et al. [2023]. Plaut and Roughgarden [2020] show that an EFX allocation exists and can be efficiently computed for non-negative additive valuations when agents assign the same ranking to all items; this result has been also extended to additive objective valuations with equally ranked items Aziz and Rey [2020]. Restricting to identical valuations, EFX allocations are known to exist for additive non-decreasing valuations Gourvès et al. [2014] and additive non-increasing valuations Barman et al. [2023], whereas they may not exist for non-monotone valuations Bérczi et al. [2024].

2 Model and Definitions

Let $N = \{1, 2, ..., n\}$ be a finite set of n agents and M be a finite set of m items. Each agent $i \in N$ has an integral valuation function $v_i: 2^M \to \mathbb{Z}$ with $v_i(\emptyset) = 0$ for any $i \in N$. We denote by $I = (N, M, (v_i)_{i \in N})$ an allocation instance. Given an agent $i \in N$, a bundle of items $S \subseteq M$ and an item $x \in S$, we say that x is a good (resp., a *chore*) for i w.r.t. S, if $v_i(S) \ge v_i(S \setminus \{x\})$ (resp., $v_i(S) \leq v_i(S \setminus \{x\})$). We observe that an item x for which $v_i(S) = v_i(S \setminus \{x\})$ is both a good and a chore for i w.r.t. S. An item x is a good (resp., a chore) if it is a good (resp., a chore) for any agent $i \in N$ w.r.t. any bundle $S \subseteq M$. An allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is a partition of M in n (possibly empty) bundles of items, such that A_i is the bundle assigned to agent $i \in N$. We aim at finding allocations satisfying fairness criteria related to envy-freeness and *equity*, as described below.

Envy-freeness. An allocation $A = (A_1, \ldots, A_n)$ is:

- envy-free (EF) if, for any $i, j \in N$, $v_i(A_i) \ge v_i(A_j)$;
- envy-free-up-to-any-item (EFX) if, for any $i, j \in N$ such that $v_i(A_i) < v_i(A_j)$, all the following conditions hold: (i) $v_i(A_i) \ge v_i(A_j \setminus \{g\})$ for any good g for agent i w.r.t. A_j ; (ii) $v_i(A_i \setminus \{c\}) \ge v_i(A_j)$ for any chore c for i w.r.t. A_i ; (iii) either there exists a good g for i w.r.t. A_j , or there exists a chore c for i w.r.t. A_i ;

- envy-free-up-to-one-item (EF1) if, for any $i, j \in N$ such that $v_i(A_i) < v_i(A_j)$, there exists $x \in A_i \cup A_j$ such that $v_i(A_i \setminus \{x\}) \ge v_i(A_j \setminus \{x\})$;
- envy-free-up-to-one-good-and-one-chore $(\mathrm{EF1}_g^c)$ if, for any $i,j \in N$ such that $v_i(A_i) < v_i(A_j)$, there exists a subset $X \subseteq M$ with $|A_i \cap X| \leq 1$ and $|A_j \cap X| \leq 1$, such that $v_i(A_i \setminus X) \geq v_i(A_j \setminus X)$.

For EF1 allocations, agent i stops envying agent j after removing at most one chore from A_i or at most one good from A_j , but not both. Similarly, under EF1_g^c allocations, agent i stops envying agent j after removing at most one chore from A_i and at most one good from A_j , even simultaneously. We observe that $\text{EF} \Rightarrow \text{EFX} \Rightarrow \text{EF1} \Rightarrow \text{EF1}_q^c$.

Equitability. The equitability notions we consider are analogous to the envy-freeness criteria described above, but the comparison each agent i makes is not against the valuation $v_i(A_i)$ that she assigns to the bundle given to any other agent j, but rather against the valuation $v_i(A_i)$ that agent j assigns to her own bundle. Thus yields allocations that are equitable (EQ), equitable-up-to-any-item (EQX), equitable-up-to-one-item (EQ1) and equitable*up-to-one-good-and-one-chore* (EQ1 $_q^c$) (see Appendix A for their explicit definition). In addition, we consider equitable-up-to-any-good-or-any-chore (EQ X_a^c) allocations, a slight relaxation of EQX that, unlike EQX, allows the equitability to hold after the removal of either goods only or chores only. In particular, an allocation \mathcal{A} is EQX_a if, for any $i \in N$, at least one of the following conditions holds: (i) for any $j \in N$ such that $v_i(A_i) < v_j(A_j)$, there exists at least one good for j w.r.t. A_i , and for any such good $g, v_i(A_i) \ge v_j(A_j \setminus \{g\})$; (ii) for any $j \in N$ such that $v_i(A_i) < v_i(A_i)$, there exists a chore for i w.r.t. A_i , and for any such chore c, $v_i(A_i \setminus \{c\}) \geq v_j(A_j)$. We observe that $EQ \Rightarrow EQX \Rightarrow EQX_q^c \Rightarrow EQ1 \Rightarrow EQ1_q^c$.

Classes of Valuations. We consider the following classes of valuations: (i) *Objective:* any item x is either a good or a chore (independently on the considered agents and bundles); in such a case, we can partition M into a set of goods G and a set of chores C (choosing arbitrarily how to classify dummy items that qualify as both); (ii) *Non-negative* (resp., *Non-positive*): $v_i(S) \geq 0$ (resp.

 $v_i(S) \leq 0$) for any $i \in N, S \subseteq M$; (iii) Monotone non-decreasing (resp., non-increasing): each item x is a good (resp., a chore), independently on the considered agents and bundles; (iv) Additive: $v_i(S) = \sum_{x \in S} v_i(x)$ for any $i, S \subseteq M$.

We will prove our positive results for the most general class of valuations, keeping in mind that monotone non-decreasing (resp. non-increasing) valuations are also non-negative (resp. non-positive), and that monotone, either non-decreasing or non-increasing, valuations are also objective; instead, objective valuations are not necessarily non-negative or non-positive, and vice versa. Finally, we assume the existence of a *constant-time oracle* that, given $i \in N$ and $S \subseteq M$, returns $v_i(S)$ in constant time.

3 Non-negative Valuations

To address the case of non-negative valuations, we consider a generalization of the fixed-point approach employed in Bilò et al. [2022], Igarashi [2023], that is extended in a non-trivial manner to handle the peculiarities of these valuations.

Fairness under Path Constraints

We say that an allocation instance is path-constrained if the m items of M are numbered from 1 to m and organized as a path P = (1, ..., m). Given $s, t \in [m] \cup \{0\}$, let $\llbracket s,t \rrbracket$ denote the bundle $\{s,s+1,\ldots,t\}$ if $t \geq s$, and the empty bundle otherwise. A bundle S is connected if S = [s, t] for some $s \in [m]$ and $t \in [m] \cup \{0\}$. An allocation A is *connected* if it is made by connected bundles only. Given a connected bundle S = [s, t], let $\partial S = \{s, t\}$ denote the *boarder* of S (observe that $\partial S = S$ if $|S| \leq 2$). For a given connected allocation $\mathcal{A} = (A_1, \dots, A_n)$, we consider the following pathbased notions of $\mathrm{EF1}_g^c$ and $\mathrm{EQ1}_g^c$: A is envy-free-up-toone-good-and-one-chore-over-paths (EF1 P_q^c) if, for any $i, j \in N$ such that $v_i(A_i) < v_i(A_j)$, there exists a subset $X \subseteq \partial A_i \cup \partial A_j$ with $|\partial A_i \cap X| \le 1$ and $|\partial A_j \cap X| \le 1$, such that $v_i(A_i \setminus X) \ge v_i(A_j \setminus X)$; \mathcal{A} is equitable-up-toone-good-and-one-chore-over-paths (EQ1P_a) if, for any $i, j \in N$ such that $v_i(A_i) < v_j(A_j)$, there exists a subset $X \subseteq \partial A_i \cup \partial A_j$ with $|\partial A_i \cap X| \leq 1$ and $|\partial A_j \cap X| \leq 1$, such that $v_i(A_i \setminus X) \ge v_j(A_j \setminus X)$.

Note that $\mathrm{EF1P}_g^c\Rightarrow\mathrm{EF1}_g^c$ and $\mathrm{EQ1P}_g^c\Rightarrow\mathrm{EQ1}_g^c$. Furthermore, the notions of $\mathrm{EQ1P}_g^c$ and $\mathrm{EF1P}_g^c$ guarantee that, even after some items are deleted, the bundles remain connected.

Sperner's Lemma.

Before presenting our results, we provide a brief overview of the underlying theoretical framework based on Sperner's Lemma. For more details, see, for example, Flegg [1974].

Let $\operatorname{conv}(\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_n)$ denote the convex hull of the n vectors $\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_n$. An (n-1)-simplex Δ is an (n-1)-dimensional polytope defined as the convex hull of its n (affinely independent) vertices $\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_n$. Given $k\in[n]$, a (k-1)-face of an (n-1)-simplex is the (k-1)-simplex obtained as convex hull of a subset of k-1 of its vertices. A triangulation T of a simplex Δ is a collection of sub-(k-1)-simplices (with $k\in[n]$) whose union is Δ , with the property that the intersection of any two sub-simplices in T is either empty or a face shared by both, which also belongs to T. Each sub-simplex $\Delta' \in T$ is referred to as an elementary simplex. The set of vertices of T, denoted as V(T), is the union of the vertices of all the elementary simplices in T (i.e., the union of all the elementary 0-simplices).

Now, let T be a fixed triangulation of an (n-1)-simplex $\Delta = \operatorname{conv}(\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n)$. A coloring function of T is a mapping $L:V(T)\to [n]$ that assigns a number, referred to as a color, from the set [n] to each vertex of T. A coloring function L is called special if, for any vertex $\boldsymbol{x}\in V(T)$ belonging to the (n-2)-face F_i of Δ that does not include \boldsymbol{v}_i (i.e., the face opposite to \boldsymbol{v}_i , obtained as convex hull of all vertices of Δ except for \boldsymbol{v}_i), the condition $L(\boldsymbol{x})\neq i$ holds. We observe that, if L is a special coloring function, then $L(\boldsymbol{v}_i)=i$ holds for any $i\in [n]$. An elementary (n-1)-simplex $\Delta^*=\operatorname{conv}(\boldsymbol{x}_1^*,\dots,\boldsymbol{x}_n^*)\in T$ is said to be fully-colored under a coloring function L if each of its n vertices is assigned a distinct color by L, that is, $L(\boldsymbol{x}_{\sigma(i)}^*)=i$ for any $i\in [n]$, for some permutation $\sigma:[n]\to[n]$.

Theorem 3.1 (Sperner's Lemma Sperner [1928]). Let T be a triangulation of an (n-1)-simplex Δ , where $n \geq 2$, and let L be a special coloring function of T. Then, there exists a fully-colored elementary (n-1)-simplex $\Delta^* \in T$

under L; moreover, the number of such simplices is odd.

See Figure 1(a) in Appendix G for an example of the application of Sperner's Lemma with n=3.

Below, we also consider a generalized version of Sperner's Lemma, as presented by Bapat [1989]. In this generalized form, there are n special coloring functions L_1,\ldots,L_n , and we seek an elementary (n-1)-simplex that is fully-colored according to a broader definition, which holds simultaneously for all of the coloring functions. Let T be a triangulation of an (n-1)-simplex, and let L_1,\ldots,L_n be the coloring functions on T. An elementary (n-1)-simplex $\Delta^*=\operatorname{conv}(\boldsymbol{x}_1^*,\boldsymbol{x}_2^*,\ldots,\boldsymbol{x}_n^*)\in T$ is jointly fully-colored under L_1,\ldots,L_n if there exist two permutations $\sigma,\tau:[n]\to[n]$ such that $L_i(\boldsymbol{x}_{\sigma(i)}^*)=\tau(i)$ for any $i\in[n]$, i.e., each vertex of Δ^* receives a distinct color under a distinct coloring function.

Theorem 3.2 (Generalized Sperner's Lemma Bapat [1989]). Let T be a triangulation of an (n-1)-simplex Δ , and let L_1, \ldots, L_n be special coloring functions of T. Then, there exists a jointly-fully-colored elementary (n-1)-simplex $\Delta^* \in T$ under L_1, \ldots, L_n .

EQ1P^c_a Allocations

Given a path-constrained allocation instance with non-negative valuations, we construct a suitable triangulation T of an n-simplex Δ and define a special coloring L for T. Each elementary (n-1)-simplex $\Delta^* \in T$ that is fully-colored under L corresponds to an $\mathrm{EQ1P}_g^c$ allocation for I. By Sperner's Lemma (Theorem 3.1), the existence of such fully-colored simplices is guaranteed, which in turn ensures the existence of an $\mathrm{EQ1P}_g^c$ allocation. We also design a polynomial-time algorithm, based on dynamic programming, that efficiently computes such an $\mathrm{EQ1P}_g^c$ allocation.

Triangulation. Consider the (n-1)-simplex $\Delta = \{x = (x_1, \dots, x_{n-1}) \in \mathbb{R}^{n-1} : 0 \leq x_1 \leq x_2 \leq \dots \leq x_{n-1} \leq m\}$, which is the convex hull $\operatorname{conv}(\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n)$ of the points $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$, with $\boldsymbol{v}_i := (0, 0, \dots, 0, m, m, \dots, m)$ for any $i \in [n]$. We observe that each of the n (n-2)-faces of Δ can be defined as $F_i := \{x = (x_1, \dots, x_{n-1}) \in \Delta : x_{i-1} = x_i\}$, where

we set $x_0:=0$ and $x_n:=m$. We construct a triangulation T of Δ whose set of vertices is $V(T)=\{x\in\Delta:x_i\in\{0,\frac{1}{3},\frac{2}{3},1,\frac{4}{3},\frac{5}{3},2,\frac{7}{3},\ldots,m-1,m-\frac{2}{3},m-\frac{1}{2},m\}\ \forall i\in[n-1]\}$ and whose simplicial structure is defined below. Each coordinate x_i of vertices $x\in V(T)$ can be either integral, or 1-fractional or 2-fractional, where integral (resp. 1-fractional, 2-fractional) means $x_i\in\mathbb{Z}$ (resp. $x_i-\frac{1}{3}\in\mathbb{Z},\ x_i-\frac{2}{3}\in\mathbb{Z}$); we write $x_i\equiv0$ (resp. $x_i\equiv1,\ x_i\equiv2$) if x_i is integral (resp. 1-fractional, 2-fractional). By leveraging Kuhn's triangulation Kuhn [1960], Scarf [1982], Deng et al. [2012], we construct the triangulation T such that each elementary (n-1)-simplex $\Delta'=\operatorname{conv}(x_1,x_2,\ldots,x_n)\in T$ can be generated by fixing the first vertex $x_1\in V(T)$ and a permitation $\pi:[n-1]\to[n-1]$, and then iteratively determining the remaining vertices as follows: $x_{i+1}=x_i+\frac{1}{3}\mathrm{e}^{\pi(i)}$ for

each $i \in [n-1]$, where $\mathbf{e}^i = (0, \dots, 0, 1, 0, \dots, 0)$ is the *i*-th vector of the canonical basis of \mathbb{R}^{n-1} . Figure 1(b) in Appendix G describes Kuhn's triangulation for n=3.

Each vertex $x \in V(T)$ can be understood as a vector representing the positions of n-1 knives that divide the interval [0,m] into n connected segments having extremes in $a,b \in [0,m] \cap \{\frac{x}{3}|x \in \mathbb{Z}\}$. Following this interpretation, the n vertices of any elementary (n-1)-simplex in T are derived by starting with an initial configuration of n-1 cuts (i.e., vertex x_1) and sequentially shifting each knife one position to the right (by a length of 1/3) according to a specific ordering defined by a permutation π . Refer to Figure 2 in Appendix G to visualize the process of deriving the sequence of fractional allocations from the n vertices of an (n-1)-dimensional simplex within a triangulation T, which is constructed from an allocation instance with n=3 agents.

Coloring Function. We now construct the coloring function $L:V(T)\to [n]$. Given a vertex $\mathbf{x}=(x_1,\ldots,x_{n-1})\in V(T)$, let $\tilde{\mathcal{A}}(\mathbf{x})=(\tilde{A}_1(\mathbf{x}),\ldots,\tilde{A}_n(\mathbf{x}))$ be the fractional connected allocation obtained from the partition of [0,m] in n fractional connected bundles, defined as $\tilde{A}_i(\mathbf{x})=[x_{i-1},x_i]$ for any $i\in [n]$, with $x_0:=0$ and $x_n:=m$; furthermore, each bundle $\tilde{A}_i(\mathbf{x})$ is assigned by default to agent i, for any $i\in [n]$ (i.e., bundles are assigned from left to right following the agents order).

Given $a \in \mathbb{R}_{\geq 0}$, let $a^- := \lfloor a \rfloor$ and $a^+ := \min\{\lfloor a \rfloor + \rfloor \}$ 1, m}. Let \tilde{v}_i denote the virtual valuation of agent i, which applies to fractional connected bundles [a, b](where $a, b \in [0, m] \cap \{\frac{x}{3} | x \in \mathbb{Z}\}$ and $a \leq b$) and returns an integer value $\tilde{v}_i([a,b])$ that is defined as follows: left-value (LV): if $a \equiv 0$, $\tilde{v}_i([a,b]) := v_i(\llbracket a^-, b^- \rrbracket)$; borderline-value (BV): if $a \equiv 1$, $\tilde{v}_i([a,b])$ is set equal to the middle value among $v_i(\llbracket a^-, b^- \rrbracket), v_i(\llbracket a^+, b^- \rrbracket)$ and $v_i(\llbracket a^+, b^+ \rrbracket)$; right-value (RV): if $a \equiv 2$, $\tilde{v}_i(\llbracket a, b \rrbracket) :=$ $v_i([a^+, b^-])$. Since the original valuations v_i s are nonnegative, the resulting virtual valuations \tilde{v}_i s are also nonnegative. Let L be the coloring function that assigns each vertex x the agent/index i that maximizes the virtual valuation $\tilde{v}_i(A_i(\boldsymbol{x}))$ applied to the fractional connected bundle $A_i(x)$, where ties are broken in favor of agents receiving a non-empty bundle and, in case of further ties, arbitrarily. We observe that L is a special coloring function. Indeed, for any $i \in [n]$, the (n-2)-face F_i of Δ , which does not contain v_i , is such that the fractional allocations $\mathcal{A}(x)$ corresponding to vertices $x \in V(T)$ located on F_i have their *i*-th bundle empty $(A_i(x) = \emptyset)$. Due to the non-negativity of the virtual valuations, any empty bundle always has the lowest virtual value, regardless of the agent or allocation being considered. Therefore, by the construction of L, we have $L(x) \neq i$ for any $i \in [n]$ and any vertex $x \in V(T)$ located on the (n-2)-face F_i . Thus, L is a special coloring function. Figure 1(b) of Appendix G shows an example of special coloring function L derived from an arbitrary non-negative virtual valuation function.

From the Fully-colored Simplex to the EQ1P $_g^c$ Allocation. According to Sperner's Lemma (Theorem 3.1), there exists at least one fully-colored elementary (n-1)-simplex $\Delta^* = \operatorname{conv}(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*) \in T$ under the coloring L, where $L(\boldsymbol{x}_{\sigma(i)}^*) = i$ for all $i \in [n]$, for some permutation $\sigma: [n] \to [n]$. Equivalently, each $i \in [n]$ is among those agents j who maximize the virtual valuation $\tilde{v}_j(\tilde{A}_j(\boldsymbol{x}_{\sigma(i)}^*))$ in the fractional connected allocation $\tilde{A}(\boldsymbol{x}_{\sigma(i)}^*)$ associated with the $\sigma(i)$ -th vertex $\boldsymbol{x}_{\sigma(i)}^*$ of Δ^* , where the sequence of allocations $\tilde{A}(\boldsymbol{x}_1^*), \dots, \tilde{A}(\boldsymbol{x}_n^*)$ is obtained by moving each knife one at a time from left to right in a specific order, starting from the position of knives determined by $\tilde{A}(\boldsymbol{x}_1^*)$. See Figure 2 of Appendix G for an example.

Denote by $\tilde{\mathcal{A}}$ the first allocation $\tilde{\mathcal{A}}(x_1^*)$, and refer to it as the main allocation of Δ^* . For a bundle $\tilde{A}_i = [a_i, b_i]$ in the main allocation, say that A_i is left-first (resp., rightfirst) in Δ^* if the first allocation $\mathcal{A}(\boldsymbol{x}_h^*)$ associated with Δ^* , for which the bundle $\tilde{A}_i(\boldsymbol{x}_h^*) = [a_i', b_i']$ differs from A_j , satisfies $a'_j = a_j + 1/3$ and $b'_j = b_j$ (resp., $a_j = a'_j$ and $b'_{i} = b_{i} + 1/3$). Equivalently, A_{j} is *left-first* (resp., right-first) in Δ^* if, among the two knives determining the endpoints of the j-th bundle across all allocations associated with Δ^* , the first to move from left to right is the left (resp., right) one; for an example of right-first bundle, see Figure 2 of Appendix G. By appropriately rounding the fractional bundles of A, we will obtain the desired (integral) allocation A that satisfies the EQ1P_q guarantee. The rounding procedure processes all fractional bundles A_j s of the main allocation \mathcal{A} from j = n down to j = 1, and for each bundle \tilde{A}_i , it returns the integral bundle A_j that will form the final (integral) allocation $\mathcal{A} = (A_1, \dots, A_n)$. Specifically, once the bundles A_{j+1}, \ldots, A_n have been determined, the bundle A_j is obtained by rounding the fractional bundle $\hat{A}_i = [a_i, b_i]$ based on the three possible fractionality levels of the two endpoints, a_j and b_j , and, if necessary, on whether \hat{A}_j is left-first or right-first. This rounding process involves considering $9 = 3 \times 3$ possible cases (corresponding to the three fractionality levels for each endpoint) and additional sub-cases, and it is formally described in Figure 3 of Appendix G. The rounding procedure that returns Ais carefully designed so that the statements of the following two lemmas hold. Their proofs, strongly based on the rounding procedure, are deferred to Appendix B.

Lemma 3.1. A is a connected (integral) allocation.

Given $i \in N$ and a connected (integral) bundle $S = \llbracket s,t \rrbracket \subseteq [m]$, let $v_i^+(S) = \max\{\llbracket s,t \rrbracket, \llbracket s+1,t \rrbracket, \llbracket s,t-1 \rrbracket\}$ and $v_i^-(S) = \min\{\llbracket s,t \rrbracket, \llbracket s+1,t \rrbracket, \llbracket s,t-1 \rrbracket\}; v_i^+(S)$ and $v_i^-(S)$ represent, respectively, the maximum and the minimum valuation that agent i can obtain from bundle S, after possibly removing one of its endpoint items.

Lemma 3.2. For any $h, i, j \in [n]$, the connected (integral) allocation \mathcal{A} satisfies $v_i^-(A_j) \leq \tilde{v}_i(\tilde{A}_j(\boldsymbol{x}_h^*)) \leq v_i^+(A_j)$.

Using these lemmas, we can show that the allocation A returned by the rounding procedure is EQ1P $_q^c$.

Theorem 3.3. A is an $EQ1P_g^c$ allocation, if valuations are non-negative.

Proof. First, \mathcal{A} is a connected allocation by Lemma 3.1. Next, we show the $\mathrm{EQ1P}_g^c$ guarantee. As observed above, the full coloring of simplex Δ^* implies that each $i \in [n]$ is one of the indices $j \in [n]$ that maximize $\tilde{v}_j(\tilde{A}_j(\boldsymbol{x}^*_{\sigma(i)}))$ (i.e., agent i has the highest virtual valuation in allocation $\tilde{\mathcal{A}}(\boldsymbol{x}^*_{\sigma(i)})$). Thus, for any $i,j \in N$, we have $v_i^+(A_i) \geq \tilde{v}_i(\tilde{A}_i(\boldsymbol{x}^*_{\sigma(i)})) \geq \tilde{v}_j(\tilde{A}_j(\boldsymbol{x}^*_{\sigma(i)})) \geq v_j^-(A_j)$, where the second inequality follows from the above observation, and the first and last inequalities follow from Lemma 3.2. Since $v_i^+(A_i) \geq v_j^-(A_j)$ for any $i,j \in N$, we conclude that \mathcal{A} satisfies the $\mathrm{EQ1P}_g^c$ guarantee (i.e., equitability is obtained by removing at most one good from the board of A_i and one chore from the board of A_j), and thus the claim holds.

Efficient Computation. The EQ1 P_q^c allocation guaranteed by Theorem 3.3 can be computed by a polynomialtime algorithm based on dynamic programming. The algorithm first computes the set C_v of the valuations $v_i(S)$ that each agent i has for any bundle S (in $O(nm^2)$ time) and then, by dynamic programming, determines for each $c \in C_v$ if there exists an allocation \mathcal{A} such that $v_i^+(A_i) \geq$ $c \geq v_i^-(A_i)$ for any $i \in [n]$ (in $O(nm^2)$ time), where $v_i^+(S)$ and $v_i^-(S)$ denote the maximum and the minimum valuation that i can obtain from a bundle S by deleting at most one item from its board; again, we restrict ourselves to allocations where the *i*-th leftmost bundle is assigned to agent i. We show that finding such a value $c \in C_v$ satisfying the above condition is equivalent to finding an $EQ1P_a^c$ allocation, whose existence is guaranteed by Theorem 3.3. Then, we get the following theorem (full details are deferred to Appendix B):

Theorem 3.4. If valuations are non-negative, an $EQ1P_g^c$ allocation can be found in time $O(n^2m^4)$.

$\mathbf{EF1P}_{q}^{c}$ allocations

To show the existence of $\mathrm{EF1P}_g^c$ allocations, we employ the same framework as in the approximate equitability case, with minor modifications. We use the same triangulation T as in the previous case but equip it with n distinct

coloring functions L_1, \ldots, L_n , instead of the single coloring function L used earlier. Here, each L_i colors any vertex in V(T) with the index j of the bundle that agent i prefers under virtual valuation \tilde{v}_i (defined as in the previous case); we note that each L_i is special, as the empty bundle is the least valuable.

By applying the Generalized Sperner's Lemma (Theorem 3.2), we show the existence of a jointly fully-colored elementary (n-1)-simplex Δ^* . As in the previous case, this simplex corresponds to a sequence of n connected fractional partitions, but now the bundles are initially unallocated, and there exist two permutations σ and τ such that, in the $\sigma(i)$ -th allocation, agent $i \in [n]$ does not envy any other agent if i receives the $\tau(i)$ -th bundle. Then, by applying the same rounding procedure and proof techniques used for the case of EQ1P $_g^c$ allocations, the first fractional partition of Δ^* is transformed into an (integral) EF1P $_g^c$ allocation, where each agent i receives the $\tau(i)$ -th bundle. This leads to the following theorem (see Appendix C for the full details).

Theorem 3.5. Under non-negative valuations, an $EF1P_g^c$ allocation always exists.

We conjecture that the computation of an EF1P $_g^c$ allocation is a PPAD-complete problem (similarly to the results show in Deng et al. [2012]). We also note that, even without path constraints, the complexity of finding EF1 or EF1 $_g^c$ still remains an open problem. It is worth noting that, in the subclass of monotone non-decreasing valuations, there are goods only. Thus, in this case, Theorems 3.3-3.5 extend to the stronger notions of EF1 and EQ1 under path-constraints, thereby recovering the findings of Bilò et al. [2022], Igarashi [2023], Misra et al. [2021], Suksompong [2019].

4 Non-positive Valuations

To address the case of non-positive valuations under path-connectivity constraints, we resort to a novel multi-coloring variant of Sperner's Lemma, where the underlying coloring functions assign, to each vertex $x \in V(T)$, a set of colors (rather than a single color), including the indices i of the (n-2)-dimensional faces F_i to which x belongs.

Multi-coloring Sperner's Lemma.

Let T be a fixed triangulation of an (n-1)-simplex $\Delta = \operatorname{conv}(\boldsymbol{v}_1, \dots, \boldsymbol{v}_n)$. A multi-coloring function of T is a mapping $\mathcal{L}:V(T) o 2^{[n]}\setminus\{\emptyset\}$ that assigns a non-empty subset of colors $\mathcal{L}(x) \subseteq [n]$ to each vertex of $x \in V(T)$. We recall that F_i is the (n-2)-dimensional face of Δ opposite to vertex v_i . A multi-coloring function \mathcal{L} is called *special* if, for any vertex $x \in V(T)$, $\mathcal{L}(x) \supseteq \{i \in [n] : x \in F_i\}$ holds (i.e., if x is a boundary vertex, the set of colors $\mathcal{L}(x)$ contains the indices associated with all (n-2)-faces of Δ on which x is located). We observe that, if \mathcal{L} is a special multi-coloring function and F is a (k-1)-face of Δ spanned by vertices v_{i_1}, \ldots, v_{i_k} , it holds that $\mathcal{L}(x) \supseteq [n] \setminus \{i_1, \ldots, i_k\}$ for any vertex $x \in V(T)$ located on F. An elementary (n-1)-simplex $\Delta^* = \operatorname{conv}(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*) \in T$ is said to be *fully-colored* under a multi-coloring function \mathcal{L} if there exists a permutation $\sigma: [n] \to [n]$ such that $i \in \mathcal{L}(\boldsymbol{x}_{\sigma(i)}^*)$ for any $i \in [n]$ (that is, a distinct color i appears in the set $\mathcal{L}(x_{\sigma(i)}^*)$ associated with a distinct vertex $x_{\sigma(i)}^*$). An example of a special multi-coloring function \mathcal{L} applied to the triangulation T of a 2-simplex is provided in Figure 4 of Appendix G.

Theorem 4.1 (Multi-coloring Sperner's Lemma). Let T be a triangulation of an (n-1)-simplex Δ , where $n \geq 2$, and let \mathcal{L} be a special multi-coloring function of T. Then, there exists a fully-colored elementary (n-1)-simplex $\Delta^* \in T$ under multi-coloring function \mathcal{L} .

The Multi-coloring Sperner's Lemma can be viewed as a dual to the standard Sperner's Lemma. In the classical version, color i is "prohibited" from appearing at any vertex x located on the face F_i opposite to the vertex v_i . In contrast, the multi-coloring version requires that color i "must appear" in the set of assigned colors $\mathcal{L}(\boldsymbol{x})$ at each vertex $\boldsymbol{x} \in F_i$. To show the Multicoloring Sperner's Lemma, we first assume, w.l.o.g., that $\mathcal{L}(\boldsymbol{x}) = \{i \in [n] : \boldsymbol{x} \in F_i\}$ holds for any vertex \boldsymbol{x} located on the boundary of Δ , and that $|\mathcal{L}(x)| = 1$ for any internal vertex x not located on the boundary. Then, we define the minimal restriction of \mathcal{L} as the standard coloring function that assigns the color $L(x) = \min\{i \in \mathcal{L}(x)\}\$ to each vertex x. We show by induction on n > 2 that the number of fully-colored simplices with respect to the minimal restriction L is odd, which implies the existence of at least

one such simplex (full details are deferred to Appendix D). Finally, we note that each fully-colored simplex with respect to L is also fully-colored with respect to the original multi-coloring function \mathcal{L} , thereby proving the claim of the theorem. The full proof is deferred to Appendix D. It is worth noting that, unlike the standard Sperner's Lemma, the structure of our multi-coloring functions required additional intermediate steps and ad hoc topological transformations of the simplicial structure in order to carry out the inductive argument on n.

$\mathbf{EQ1P}_g^c$ and $\mathbf{EF1P}_g^c$ Allocations

To show the existence of an EQ1P $_q^c$ allocation, we employ the same framework as in the case of non-negative valuations, with minor modifications. We use the same triangulation T as in that case, but we equip T with the multicoloring function \mathcal{L} that assign to each vertex $x \in V(T)$ the set $\mathcal{L}(x)$ of indices j which maximize $\tilde{v}_i(\tilde{A}_i(x))$, where \tilde{v}_i is the virtual valuation defined as in Section 3. Differently from the case of non-negative valuations, under non-positive valuations the empty bundle is always the best one for each agent. Thus, \mathcal{L} is a special multicoloring function and, by the Multi-coloring Sperner's lemma (Theorem 4.1), there exists an elementary (n-1)simplex $\Delta^* = \text{conv}(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*)$ that is fully-colored under the multi-coloring function \mathcal{L} . As in the case of nonnegative valuations, this means that there exists a permutation σ such that each agent $i \in [n]$ is the happiest (among all others) in the allocation $\mathcal{A}(x^*_{\sigma(i)})$ (where each agent j receives the j-th bundle). From this point onward, we can apply the same approach used for non-negative valuations to transform Δ^* into an EQ1P^c_a allocation (full details are deferred to Appendix D).

Theorem 4.2. Under non-positive valuations, an $EQ1P_g^c$ allocation always exists and can be computed in polynomial time.

As in the case of non-negative valuations, to show the existence of $EF1P_g^c$ allocations, we consider a generalization of the multi-coloring Sperner's Lemma that deals with n distinct multi-coloring functions, each one modelling the virtual valuation of each agent i. We then obtain a jointly fully-colored simplex Δ^* , in which each vertex corresponds to an allocation where a distinct agent prefers a distinct bundle. Then, by resorting to the usual rounding

procedure we obtain the desired $EF1P_g^c$ allocation (full details are deferred to Appendix D):

Theorem 4.3. Under non-positive valuations, an $EF1P_g^c$ allocation always exists.

The following corollary holds since, in the case of monotone non-increasing valuations, there are chores only.

Corollary 4.1. Under non-increasing valuations, EQ1 and EF1 allocations always exist, even under path constraints, with the former being computable in polynomial time

5 Conclusions

As the main contribution of this work, we established the existence of allocations that are (approximate) equitable $(EQ1_q^c)$ or envy-free $(EF1_q^c)$ up to the removal of one good or chore from each bundle, even for non-positive valuations and under path constraints. Furthermore, efficient computation can be achieved under the approximate equitability guarantee. With these results, we made significant progress on the general problem of establishing the existence and computation of allocations that are fair "up to some items" for general non-monotone valuations. However, the existence of allocations satisfying the stronger EQ1 and EF1 guarantees remains open, as does the case of more general non-monotone valuations beyond the settings of the non-negative or non-positive valuations. It would be interesting to explore whether our techniques can be extended to address these cases.

Additionally, the time complexity of finding approximately fair allocations for general non-negative or non-positive valuations remains an open question, even in cases where existence has been established. Indeed, under path constraints, we conjecture that finding EF1P_g^c allocations is PPAD-complete for both non-negative and non-positive valuations. Furthermore, we conjecture that PPAD-completeness also holds for finding EF1 allocations under monotone non-decreasing or non-increasing valuations.

Finally, the existence of EQX allocations for objective valuations remains an open problem, even in the additive case. Furthermore, moving to the approximate envyfreeness guarantee, we note that the existence of EFX allocations is a major open question in fair division, even for additive non-negative valuations.

AI Use Declaration

ChatGPT (OpenAI) was used solely for language polishing and figure layout formatting. All scientific content and data were prepared by the authors.

References

- 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. *Artif. Intell.*, 322:103965, 2023.
- Yonatan Aumann and Yair Dombb. The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation*, 3(4):1–16, 2015.
- Haris Aziz and Simon Rey. Almost group envy-free allocation of indivisible goods and chores. In *Proceedings* of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pages 39–45, 2020.
- Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. *Auton. Agents Multi Agent Syst.*, 36(1):3, 2022.
- R. B. Bapat. A constructive proof of a permutation-based generalization of Sperner's lemma. *Mathematical Programming*, 44(1):113–120, 1989.
- Siddharth Barman, Vishnu V. Narayan, and Paritosh Verma. Fair chore division under binary supermodular costs. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems AAMAS*, pages 2863–2865. ACM, 2023.
- Siddharth Barman, Umang Bhaskar, Yeshwant Pandit, and Soumyajit Pyne. Nearly equitable allocations beyond additivity and monotonicity. In *38th AAAI Conference on Artificial Intelligence, AAAI 2024*, pages 9494–9501. AAAI Press, 2024.

- Kristóf Bérczi, Erika R. Bérczi-Kovács, Endre Boros, Fekadu Tolessa Gedefa, Naoyuki Kamiyama, Telikepalli Kavitha, Yusuke Kobayashi, and Kazuhisa Makino. Envy-free relaxations for goods, chores, and mixed items. *Theor. Comput. Sci.*, 1002:114596, 2024.
- Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. On approximate envy-freeness for indivisible chores and mixed resources. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM, volume 207 of LIPIcs, pages 1:1–1:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 2021.
- Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. Connected equitable cake division via sperner's lemma. *Inf. Process. Lett.*, 189:106554, 2025.
- Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S. Zwicker. Almost envyfree allocations with connected bundles. *Games Econ. Behav.*, 131:197–221, 2022.
- Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 135–141, 2017.
- Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. Chore division on a graph. *Auton. Agents Multi Agent Syst.*, 33(5):540–563, 2019.
- Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
- 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.
- Katarína Cechlárová, Jozef Doboš, and Eva Pillárová. On the existence of equitable cake divisions. *Information Sciences*, 228:239–245, 2013.

- Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems* 30: Annual Conference on Neural Information Processing Systems 2017, pages 5029–5037, 2017.
- Guillaume Chèze. Existence of a simple and equitable fair division: A short proof. *Mathematical Social Sciences*, 87:92–93, 2017.
- X. Deng, Q. Qi, and A. Saberi. Algorithmic solutions for envy-free cake cutting. *Operations Research*, 60 (6):1461–1476, 2012.
- Michael Dinitz, Aravind Srinivasan, Leonidas Tsepenekas, and Anil Vullikanti. Fair disaster containment via graph-cut problems. In *International Conference on Artificial Intelligence and Statistics, AISTATS* 2022, 28-30 March 2022, Virtual Event, volume 151 of Proceedings of Machine Learning Research, pages 6321–6333. PMLR, 2022.
- Lester E. Dubins and Edwin H. Spanier. How to cut a cake fairly. *American Mathematical Monthly*, 68(1P1): 1–17, 1961.
- H. Graham Flegg. From Geometry to Topology. Crane, Russak & Co., 1974.
- Duncan K. Foley. *Resource allocation and the public sector*. Yale University, 1966.
- Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable allocations of indivisible goods. In 28th International Joint Conference on Artificial Intelligence, IJCAI 2019, pages 280–286, 2019.
- Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable allocations of indivisible chores. In 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2020, pages 384–392, 2020.
- Jugal Garg and Aniket Murhekar. Computing paretooptimal and almost envy-free allocations of indivisible goods. *J. Artif. Intell. Res.*, 80:1–25, 2024.
- Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. Near fairness in matroids. In 21st European Conference

- on Artificial Intelligence, ECAI 2014, pages 393–398, 2014.
- Hadi Hosseini and Aditi Sethia. Equitable allocations of mixtures of goods and chores. *CoRR*, abs/2501.06799, 2025.
- Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Fairly dividing mixtures of goods and chores under lexicographic preferences. In *Proceedings of the* 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, pages 152–160. ACM, 2023.
- Ayumi Igarashi. How to cut a discrete cake fairly. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, pages 5681–5688. AAAI Press, 2023.
- H. W. Kuhn. Some combinatorial lemmas in topology. *IBM Journal of Research and Development*, 4(5):518–524, 1960.
- 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 2004*, pages 125–131, 2004.
- Shengxin Liu, Xinhang Lu, Mashbat Suzuki, and Toby Walsh. Mixed fair division: A survey. *J. Artif. Intell. Res.*, 80:1373–1406, 2024.
- Neeldhara Misra, Chinmay Sonar, P. R. Vaidyanathan, and Rohit Vaish. Equitable division of a path. *CoRR*, abs/2101.09794, 2021.
- Benjamin Plaut and Tim Roughgarden. Almost envyfreeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2):1039–1068, 2020.
- H. E. Scarf. The computation of equilibrium prices: an exposition. *Handbook of Mathematical Economics*, 2: 1007–1061, 1982.
- Roy Schwartz and Roded Zats. Fair correlation clustering in general graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022*, volume 245 of *LIPIcs*, pages 37:1–37:19, 2022.

- Emanuel Sperner. Neuer beweis für die invarianz der dimensionszahl und des gebietes. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 6: 265–272, 1928.
- Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948.
- Walter Stromquist. How to cut a cake fairly. *American Mathematical Monthly*, 87(8):640–644, 1980.
- Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly*, 106 (10):936–942, 1999.
- Warut Suksompong. Fairly allocating contiguous blocks of indivisible items. *Discret. Appl. Math.*, 260:227–236, 2019.
- Douglas Robert Woodall. Dividing a cake fairly. *Journal of Mathematical Analysis and Applications*, 78(1):233–247, 1980.

Supplementary Material of Paper "Approximately Envy-free and Equitable Allocations of Indivisible Items for Non-monotone Valuations" and Reproducibility Checklist

The supplementary material is organized in seven distinct appendices (A-H). For a better organization, each distinct appendix starts in a new page. The reproducibility checklist is provided at the end of the document.

A Explicit Description of the Approximate Equitability Notions (from Section 2)

An allocation $\mathcal{A} = (A_1, \dots, A_n)$ is:

- equitable (EQ): if, for any $i, j \in N$, $v_i(A_i) \ge v_j(A_j)$.
- equitable-up-to-any-item (EQX): if, for any $i, j \in N$ such that $v_i(A_i) < v_j(A_j)$, all the following conditions hold: (i) $v_i(A_i) \ge v_j(A_j \setminus \{g\})$ for any good g for j w.r.t. A_j ; (ii) $v_i(A_i \setminus \{c\}) \ge v_j(A_j)$ for any chore c for i w.r.t. A_i ; (iii) either there exists a good g for j w.r.t. A_j , or there exists a chore c for i w.r.t. A_i .
- equitable-up-to-one-item (EQ1): if, for any $i, j \in N$ such that $v_i(A_i) < v_j(A_j)$, there exists $x \in A_i \cup A_j$ such that $v_i(A_i \setminus \{x\}) \ge v_j(A_j \setminus \{x\})$.
- envy-free-up-to-one-good-and-one-chore (EQ1 $_g^c$): if, for any $i, j \in N$ such that $v_i(A_i) < v_j(A_j)$, there exists a subset $X \subseteq M$ with $|A_i \cap X| \le 1$ and $|A_j \cap X| \le 1$, such that $v_i(A_i \setminus X) \ge v_j(A_j \setminus X)$.

B Missing Proofs from Section 3: part on equitability

Proof of Lemma 3.1

The rounding procedure described in Figure 3 of Appendix G verifies the following properties: (i) the right endpoint t_n of bundle $A_n = \llbracket s_n, t_n \rrbracket$ is m; (ii) for any $j \in [n-1]$, the right endpoint t_j of bundle $A_j = \llbracket s_j, t_j \rrbracket$ and the left endpoint s_{j+1} of bundle A_{j+1} satisfy $t_j = s_{j+1}-1$; (iii) the left endpoint s_1 of bundle A_1 is 1. Based on these observations, all the connected bundles A_j are disjoint and their union covers all items in M.

Proof of Lemma 3.2

For the rounding procedure, refer to Figure 3 in Appendix G. Fix $h, i, j \in [n]$ and let \mathcal{A}' denote allocation $\mathcal{A}(\mathbf{x}_h^*)$. Let $A_j(L, L) := [a_j, b_j], A_j(L, R) := [a_j, \min\{b_j + a_j\}]$ 1/3, m], $\tilde{A}_{j}(R, L) := [\min\{a_{j} + 1/3, b_{j}\}, b_{j}],$ and $\tilde{A}_i(R,R) := [\min\{a_j + 1/3, b_j\}, \min\{b_j + 1/3, m\}].$ Equivalently: $\tilde{A}_{j}(L, L)$ represents the j-th fractional bundle \tilde{A}_i of the allocation $\tilde{\mathcal{A}} = \tilde{\mathcal{A}}(\boldsymbol{x}^*)$ associated with the first vertex of Δ^* ; $\hat{A}_i(L,R)$ is the fractional bundle obtained from \tilde{A}_i by moving the right knife one position (i.e., 1/3 of the size of an item) to the right, ensuring it remains before the last possible position; $\tilde{A}_i(R,L)$ is the fractional bundle obtained by moving the left knife one position to the right (while keeping it before the right knife); $A_i(R,R)$ is the fractional bundle obtained by moving both knives one position to the right, ensuring the left knife remains before the right knife. We observe that each possible configuration of the j-th bundle in \hat{A} must be one of $\hat{A}_i(L,L)$, $\hat{A}_i(L,R)$, $\hat{A}_i(R,L)$, and $A_i(R,R)$. Thus, to obtain the claim, it is sufficient to show that $v_i^-(A_i) \leq \tilde{v}_i(\tilde{A}_i(Y,Z)) \leq v_i^+(A_i)$ for any $Y,Z \in \{L,R\}.$

At this point, the remainder of the proof is straightforward but tedious, as it requires verifying the claim by systematically analyzing all nine cases and their respective sub-cases in the rounding procedure. In particular, for each of the nine cases, one must verify that inequality $v_i^-(A_j) \leq \tilde{v}_i(\tilde{A}_j(Y,Z)) \leq v_i^+(A_j)$ holds for any $Y,Z \in \{L,R\}$.

As an example, we describe the analysis of cases 2(i), 2(ii), 6(i) and 6(ii), since the other cases are less complex and can be automatically verified by following analogous and simpler arguments.

We recall that the virtual value $\tilde{v}_i([a,b])$ of a fractional bundle [a,b] corresponds to one of the following three cases:

- *left-value (LV)*: if $a \equiv 0$, $\tilde{v}_i([a,b]) := v_i([a^-,b^-])$;
- borderline-value (BV): if $a \equiv 1$, $\tilde{v}_i([a,b])$ is set equal to the middle value among $v_i([a^-,b^-])$, $v_i([a^+,b^-])$ and $v_i([a^+,b^+])$;
- right-value (RV): if $a \equiv 2$, $\tilde{v}_i([a,b]) := v_i(\llbracket a^+, b^- \rrbracket)$.

Furthermore, we recall that the j-th bundle \tilde{A}_j of the main allocation $\tilde{\mathcal{A}}$ is *left-first* (resp., *right-first*) in Δ^* if, among the two knives determining the endpoints of the j-th bundle across all allocations associated with Δ^* , the first to move from left to right is the left (resp., right) one.

In the following, we proceed with the analysis of Cases 2(i), 2(ii), 6(i) and 6(ii):

• Case 2(i): In this case, we have $b_j \in A_{j+1}$ and we set $A_j \leftarrow [\![a_j^-, b_j^-]\!]$.

First, let us focus on fractional bundle $\tilde{A}_j(L,Z)$ for any $Z \in \{L,R\}$; we observe that the virtual valuation applied to $\tilde{A}_j(L,Z)$ is left-value (LV), that is, $\tilde{v}_i(\tilde{A}_j(L,Z))$ is equal to $v_i(\llbracket a_j^-,b_j^- \rrbracket)$. Thus, we have $v_i^-(A_j) \leq v_i(\llbracket a_j^-,b_j^- \rrbracket) = \tilde{v}_i(\tilde{A}_j(L,Z)) = v_i(\llbracket a_j^-,b_j^- \rrbracket) \leq v_i^+(A_j)$, where the first (resp., the last) inequality holds since $v_i^-(A_j)$ (resp., $v_i^+(A_j)$) is the highest (resp., lowest) valuation obtainable by removing at most one boundary item, or none at all, from A_j .

Now, we consider a fractional bundle of type $\tilde{A}_j(R,Z)$ with $Z\in\{L,R\}$; the virtual valuation applied to $\tilde{A}_j(R,Z)$ is borderline-value (BV), that is, $\tilde{v}_i(\tilde{A}_j(R,Z))$ is equal to the middle value among $v_i(\llbracket a_j^-,b_j^- \rrbracket)$, $v_i(\llbracket a_j^+,b_j^- \rrbracket)$ and $v_i(\llbracket a_j^+,b_j^+ \rrbracket)$. Thus, we have $v_i^-(A_j) \leq \min\{v_i(\llbracket a_j^-,b_j^- \rrbracket),v_i(\llbracket a_j^+,b_j^- \rrbracket)\} \leq \tilde{v}_i(\tilde{A}_j(R,Z)) \leq \max\{v_i(\llbracket a_j^-,b_j^- \rrbracket),v_i(\llbracket a_j^+,b_j^- \rrbracket)\} \leq$

 $v_i^+(A_j)$, where: the first (resp., the last) inequality holds again since $v_i^-(A_j)$ (resp., $v_i^+(A_j)$) is the highest (resp., lowest) valuation obtainable by removing at most one boundary item, or none at all, from A_j ; the second (resp., third) inequality holds since the minimum (resp., maximum) among the two values $v_i((a_j^-,b_j^-))$ and $v_i((a_j^+,b_j^-))$ must be less than or equal to (resp., greater than or equal to) the middle value among the three values $v_i((a_j^-,b_j^-))$, $v_i((a_j^+,b_j^-))$ and $v_i((a_j^+,b_j^+))$.

We conclude that $v_i^-(A_j) \leq \tilde{v}_i(\tilde{A}_j(Y,Z)) \leq v_i^+(A_j)$ holds for any $Y,Z \in \{L,R\}$, and this shows the claim in case 2(i).

Case 2(ii): In this case, we have b_j ∉ A_{j+1} and we set A_j = [[a_i⁻, b_i⁺]].

First, let us focus on fractional bundle $\tilde{A}_j(L,Z)$ for any $Z\in\{L,R\}$; we observe that the virtual valuation applied to $\tilde{A}_j(L,Z)$ is left-value (LV), that is, $\tilde{v}_i(\tilde{A}_j(L,Z))$ is equal to $v_i(\llbracket a_j^-,b_j^- \rrbracket)$. Thus, we have $v_i^-(A_j) \leq \min\{v_i(\llbracket a_j^-,b_j^+ \rrbracket),v_i(\llbracket a_j^-,b_j^- \rrbracket)\} \leq \tilde{v}_i(\tilde{A}_j(L,Z)) \leq \max\{v_i(\llbracket a_j^-,b_j^+ \rrbracket),v_i(\llbracket a_j^-,b_j^- \rrbracket)\} \leq v_i^+(A_j)$, where the first and the last inequality holds for the usual reasons, while the second and third inequalities hold since $\tilde{v}_i(\tilde{A}_j(L,Z)) = v_i(\llbracket a_j^-,b_j^- \rrbracket)$.

Now, we consider each fractional bundle $A_i(R,Z)$ with $Z \in \{L,R\}$; the virtual valuation applied to $A_i(R,Z)$ is borderlinevalue (BV), that is, $\tilde{v}_i(A_i(R,Z))$ is equal to the middle value among $v_i(\llbracket a_i^-, b_i^- \rrbracket)$, $v_i(\llbracket a_i^+, b_i^- \rrbracket)$ and $v_i(\llbracket a_i^+, b_i^+ \rrbracket)$. Thus, we have $v_i^-(A_j) \leq \min\{v_i([[a_i^-, b_i^-]]), v_i([[a_i^+, b_i^+]])\} \leq$ $\tilde{v}_i(\tilde{A}_j(R,Z)) \le \max\{v_i([[a_i^-, b_i^-]]), v_i([[a_i^+, b_i^+]])\} \le$ $v_i^+(A_i)$, where the first and the last inequality holds for the usual reasons, while, similarly to case 2(i), the second (resp., third) inequality holds since the minimum (resp., maximum) among the two values $v_i((a_j^-, b_j^-))$ and $v_i((a_j^+, b_j^+))$ must be less than or equal to (resp., greater than or equal to) the middle value among the three values $v_i((a_i^-, b_i^-))$, $v_i((a_i^+, b_i^-))$ and $v_i((a_i^+, b_i^+))$.

We conclude that $v_i^-(A_j) \leq \tilde{v}_i(\tilde{A}_j(Y,Z)) \leq$

 $v_i^+(A_j)$ holds for any $Y,Z\in\{L,R\}$, and this shows the claim in case 2(ii).

• Case 6(i): In this case, we have $A_j = [a_j^+, b_j^+]$. As the fractional bundle $[a_j, b_j]$ is left-first in Δ^* , the j-th bundle will never be equal to $\tilde{A}_j(L,R)$ over all allocations associated with vertices in Δ^* . Thus, it is sufficient showing that $v_i^-(A_j) \leq \tilde{v}_i(\tilde{A}_j(Y,Z)) \leq v_i^+(A_j)$ holds for any $(Y,Z) \in \{(L,L),(R,L),(R,R)\}$.

First, let us focus on fractional bundle $\tilde{A}_j(L,L)$; the virtual valuation applied to $\tilde{A}_j(L,L)$ is borderline-value (BV), that is, $\tilde{v}_i(\tilde{A}_j(L,L))$ is equal to the middle value among $v_i(\llbracket a_j^-,b_j^-\rrbracket),v_i(\llbracket a_j^+,b_j^-\rrbracket)$ and $v_i(\llbracket a_j^+,b_j^+\rrbracket)$. Thus, for the usual reasons, we have $v_i^-(A_j) \leq \min\{v_i(\llbracket a_j^+,b_j^-\rrbracket),v_i(\llbracket a_j^+,b_j^+\rrbracket)\} \leq \tilde{v}_i(\tilde{A}_j(L,L)) \leq \max\{v_i(\llbracket a_j^+,b_j^-\rrbracket),v_i(\llbracket a_j^+,b_j^+\rrbracket)\} \leq v_i^+(A_j)$.

 $A_i(R,L);$ Now, we consider bundle valuation applied $A_i(R,L)$ is virtual to (RV), that $\tilde{v}_i(A_i(R,Z))$ is, is equal to $v_i(\llbracket a_j^+, b_j^- \rrbracket)$. Thus, we have $v_i^-(A_j) \leq \min\{v_i([[a_j^+, b_j^-]]), v_i([[a_j^+, b_j^+]])\} \leq$ $\tilde{v}_i(\tilde{A}_j(R,Z)) \le \max\{v_i([[a_j^+,b_j^-]]),v_i([[a_j^+,b_j^+]])\} \le$ $v_i^+(A_i)$, where the first and the second inequalities hold for the usual reasons, and the other inequalities hold since $\tilde{v}_i(A_j(R,Z)) = v_i(\llbracket a_i^+, b_i^- \rrbracket)$.

Finally, we consider bundle $A_j(R,R)$; the virtual valuation applied to $\tilde{A}_j(R,R)$ is again right-value (RV), but now the right knife has position b^+ , and then $\tilde{v}_i(\tilde{A}_j(R,R)) = v_i(\llbracket a_j^+,b_j^+ \rrbracket)$. Thus, for the usual reasons, we have $v_i^-(A_j) \leq v_i(\llbracket a_j^+,b_j^+ \rrbracket) = \tilde{v}_i(\tilde{A}_j(R,R)) = v_i(\llbracket a_j^+,b_j^+ \rrbracket) \leq v_i^+(A_j)$.

We conclude that $v_i^-(A_j) \leq \tilde{v}_i(\tilde{A}_j(Y,Z)) \leq v_i^+(A_j)$ holds for any $(Y,Z) \in \{(L,L),(R,L),(R,R)\}$, and this shows the claim in case 6(i).

• Case 6(ii): In this case, we have $A_j = [\![a_j^-, b_j^+]\!]$. As the fractional bundle $[a_j, b_j]$ is right-first in Δ^* , the j-th bundle will never be equal to $\tilde{A}_j(R, L)$ over all allocations associated with vertices in Δ^* . Thus, it is sufficient showing that $v_i^-(A_j) \leq$

 $\tilde{v}_i(\tilde{A}_j(Y,Z)) \leq v_i^+(A_j)$ holds for any $(Y,Z) \in \{(L,L),(L,R),(R,R)\}.$

First, let us focus on fractional bundle $\tilde{A}_j(L,L)$; the virtual valuation applied to $\tilde{A}_j(L,L)$ is borderline-value (BV), that is, $\tilde{v}_i(\tilde{A}_j(L,L))$ is equal to the middle value among $v_i(\llbracket a_j^-,b_j^- \rrbracket),v_i(\llbracket a_j^+,b_j^- \rrbracket)$ and $v_i(\llbracket a_j^+,b_j^+ \rrbracket)$. Thus, for the usual reasons, we have $v_i^-(A_j) \leq \min\{v_i(\llbracket a_j^-,b_j^- \rrbracket),v_i(\llbracket a_j^+,b_j^+ \rrbracket)\} \leq \tilde{v}_i(\tilde{A}_j(L,L)) \leq \max\{v_i(\llbracket a_j^-,b_j^- \rrbracket),v_i(\llbracket a_j^+,b_j^+ \rrbracket)\} \leq v_i^+(A_j).$

Now, we consider bundle $\tilde{A}_j(L,R)$; the virtual valuation applied to $\tilde{A}_j(R,L)$ is again borderline-value (BV), but the right knife is now placed in position b^+ , that is, $\tilde{v}_i(\tilde{A}_j(L,R))$ is equal to the middle value among $v_i(\llbracket a_j^-,b_j^+ \rrbracket)$, $v_i(\llbracket a_j^+,b_j^+ \rrbracket)$ and $v_i(\llbracket a_j^+,(b_j^-+1)^+ \rrbracket)$. Thus, for the usual reasons, we have $v_i^-(A_j) \leq \min\{v_i(\llbracket a_j^-,b_j^+ \rrbracket),v_i(\llbracket a_j^+,b_j^+ \rrbracket)\} \leq \tilde{v}_i(\tilde{A}_j(L,L)) \leq \max\{v_i(\llbracket a_j^-,b_j^+ \rrbracket),v_i(\llbracket a_j^+,b_j^+ \rrbracket)\} \leq v_i^+(A_j)$.

Finally, we consider bundle $\tilde{A}_j(R,R)$; the virtual valuation applied to $\tilde{A}_j(R,L)$ is right-value (RV) and the right knife is again in position b^+ , that is, $\tilde{v}_i(\tilde{A}_j(R,R)) = v_i([a_j^+,b_j^+])$. Thus, for the usual reasons, we have $v_i^-(A_j) \leq \min\{v_i([a_j^-,b_j^+]),v_i([a_j^+,b_j^+])\} \leq v_i(\tilde{A}_j(R,R)) \leq \max\{v_i([a_j^-,b_j^+]),v_i([a_j^+,b_j^+])\} \leq v_i(\tilde{A}_j(R,R))$. Theorem B.1

We conclude that $v_i^-(A_j) \leq \tilde{v}_i(\tilde{A}_j(Y,Z)) \leq v_i^+(A_j)$ holds for any $(Y,Z) \in \{(L,L),(L,R),(R,R)\}$, and this shows the claim in case 6(ii).

Full Details on Efficient Computation of EQ1P $_g^c$ Allocations (Theorem B.1).

Full details on design, correctness and complexity of the dynamic programming algorithm introduced in Section 3 are provided in the proof of Theorem B.1 below (for the pseudocode, see Algorithm 1 of Appendix H).

We say that a connected allocation A is well-ordered if the i-th leftmost bundle is assigned to agent i, for any

 $i \in [n]$; we note the allocation \mathcal{A} constructed in Theorem 3.3 to show the existence of EQ1P $_g^c$ allocations was well-ordered. In this section, we again restrict our attention to well-ordered allocations, and this assumption is without loss of generality (see Remark B.1). Let C_v be the set of the valuations $v_i(S)$ that each agent i has for any bundle S, and, given $c \in C_v$, we say that an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is c-feasible if, for each $i \in [n]$, $v_i^+(A_i(c)) \geq c \geq v_i^-(A_i(c))$ holds.

Lemma B.1. A is a well-ordered c-feasible allocation for some $c \in C_v \Leftrightarrow A$ is a well-ordered $EQ1P_q^c$ allocation.

Proof. \Rightarrow): Let $\mathcal A$ be a c-feasible well-ordered allocation for some $c \in C_v$. Then, $v_i^+(A_i) \geq c \geq v_j^-(A_j)$ holds for any $i,j \in N$, that is, $\mathcal A$ is $\mathrm{EQ1P}_q^c$.

 $\Leftarrow) \text{: Given an EQ1P}_g^c \text{ well-ordered allocation } \mathcal{A}, \text{ let } c^+ := \min_{j \in N} v_j^+(A_j) \text{ and } c^- := \max_{j \in N} v_j^-(A_j).$ Then, we have that $v_i^+(A_i) \geq c^+ \geq c^- \geq v_i^-(A_i)$ holds for any $i \in N$, where the intermediate inequality follows from the EQ1P $_g^c$ guarantee. Let j be the index minimizing $v_j^+(A_j)$, that is, $c^+ = v_j^+(A_j)$. We observe that $v_j^+(A_j) = v_j(S)$, for some bundle S obtained from A_j by possibly removing one item from the board of A_j . Thus, by setting $c := c^+$, we have $c \in C_v$. Then, by the previous inequalities, we have $v_i^+(A_i) \geq c \geq v_i^-(A_i)$ for any $i \in [n]$, that is, \mathcal{A} is a well-ordered c-feasible allocation with $c \in C_v$.

Then, we obtain Theorem 3.4, that is restated and proven below:

Theorem B.1 (Claim of Theorem 3.4). Under non-negative valuations, an $EQ1P_g^c$ allocation can be found in polynomial time $O(n^2m^4)$.

Proof. By Lemma B.1, to find an EQ1P_g^c allocation \mathcal{A} , it is sufficient to check, for each $c \in C_v$, whether there exists a well-ordered c-feasible allocation. Furthermore, the existence of a well-ordered EQ1P_g^c allocation was guaranteed by Theorem 3.3 (whose proof, indeed, yielded a well-ordered allocation), so it is certain that such a $c \in C_v$ can be found.

For any $c \in C_v$, the problem of determining whether there exists a well-ordered c-feasible allocation and, if so, finding such an allocation can be solved using a dynamic programming algorithm in polynomial time $O(nm^2)$. Indeed, one can iteratively solve the sub-problem $B_c[i][j]$

which determines whether there exists a partial well-ordered c-feasible allocation that allocates the first j items to the first i agents.

Specifically, for any $i \in [n]$ and $j \in [m] \cup \{0\}$, let $B_c[i][j]$ be the boolean value that is TRUE (resp. FALSE) if there exists a (resp. if there is no) partial c-feasible well-ordered allocation $\mathcal A$ of the first j items to the first i agents. The solution to the original problem is then obtained by evaluating $B_c[n][m]$. The values $B_c[i][j]$ satisfy the following recurrence relation:

- Base Case $(i = 1, j \in [m] \cup \{0\})$: $B_c[1][j] = \text{TRUE}$ if the bundle [1, j] satisfies $v_1^+([1, j]) \ge c \ge v_1^-([1, j])$; otherwise, $B_c[1][j] = \text{FALSE}$.
- Recursive Step $(i \in [n] \setminus \{1\}, j \in [m] \cup \{0\})$: $B_c[i][j] = \text{TRUE}$ if there exists $\ell \in [j+1]$ such that the bundle $[\![\ell,j]\!]$ satisfies both $v_i^+([\![\ell,j]\!]) \geq c \geq v_i^-([\![\ell,j]\!])$ and $B_c[i-1][\ell-1] = \text{TRUE}$; otherwise, $B_c[i][j] = \text{FALSE}$.

The above recurrence relation for $B_c[i][j]$ can be exploited by a dynamic-programming algorithm to compute all the values $B_c[i][j]$, for $i \in [n]$ and $j \in [m] \cup \{0\}$, in $O(nm^2)$ time (i.e., O(m) to compute each cell $B_c[i][j]$, multiplied by the total number of cells, $O(mn))^1$. Thus, the overall execution of this dynamic-programming approach for all $c \in C_v$ requires $O((nm^2) \cdot (nm^2)) = O(n^2m^4)$ steps.

A pseudo-code of the procedure outlined above is given in Algorithm 1 of Appendix H. \Box

We observe that, under identical valuations, the cardinality of the set C_v is $O(m^2)$, as the value of each connected bundle does not depend on the agent evaluating it. In this case, the complexity of the algorithm outlined in the proof of Theorem 3.4 decreases by a factor of n. Moreover, since all the considered approximate envyfreeness and equitability guarantees coincide under identical valuations, we obtain the following corollary as a direct consequence of Theorems 3.3 and B.1 (restated above as Theorem B.1).

Corollary B.1. Under identical non-negative valuations, $EQ1P_g^c$ (or, equivalently, $EF1P_g^c$) allocations always exist and can be efficiently computed in polynomial time $O(n^2m^3)$.

Remark B.1. We observe that the $EQ1P_g^c$ allocations \mathcal{A} considered in Theorem 3.4 (as well as that of Theorem 3.3) is well-ordered, that is, it assigns the left-most bundle to agent 1, the second left-most bundle to agent 2, and so forth. Anyway, the existence and computation of $EQ1P_g^c$ allocations can be easily generalized to accommodate any fixed ordering of the agents, that is, we can decide in advance which agent is assigned the i-th left-most bundle. Indeed, by Theorem 3.4, for any fixed ordering $\rho:[n] \to [n]$ of the agents, we can compute in polynomial time another $EQ1P_g^c$ allocation that is well-ordered w.r.t. the new agents order, that is, it assigns the left-most bundle to agent $\rho(1)$, the second left-most bundle to agent $\rho(2)$, and so forth.

 $^{^1}$ For i=n, it is sufficient to find the value $B_c[n][j]$ for j=m only, since the bundle assigned to agent n in the complete allocation is either empty or must include the last item m. Nonetheless, for the sake of simplicity, we compute all values $B_c[n][j]$, as this does not impact the asymptotic time complexity.

C Missing Proofs from Section 3: part on envy-freeness

Full details on the proof of Theorem 3.5

To show the existence of an EF1 $_g^c$ allocation, we employ the same framework as in the approximate equitability case, with minor modifications. Specifically, we use the same triangulation T as in the previous case but equip it with n distinct coloring functions L_1, \ldots, L_n , instead of the single coloring function L used earlier.

Here, each vertex $\mathbf{x} \in V(T)$ corresponds to a partition $\tilde{\mathcal{S}}(\mathbf{x}) = (\tilde{S}_1(\mathbf{x}), \dots, \tilde{S}_n(\mathbf{x}))$ of the interval [0, m], referred to as a fractional connected partition. This partition divides the interval into n connected sub-intervals (fractional connected bundles), arranged sequentially from the left-most to the right-most. Each fractional bundle is temporarily unassigned to any agent, unlike in the equitability case (where the i-th left-most fractional bundle was automatically assigned to agent i).

Let \tilde{v}_i be the virtual valuation defined as in the equitability case. For any $i \in [n]$, consider a distinct coloring function L_i that assigns to each vertex $\boldsymbol{x} \in V(T)$ the index j of the bundle $\tilde{S}_j(\boldsymbol{x})$ in $\tilde{\mathcal{S}}(\boldsymbol{x})$ that maximizes the virtual valuation $\tilde{v}_i(\tilde{S}_j(\boldsymbol{x}))$ of agent i, with ties broken in favor of non-empty bundles and, in case of further ties, arbitrarily. Also in this case, each L_i is a special coloring function (since the empty bundle is the least valuable for any agent).

By the Generalized Sperner's Lemma (Theorem 3.2), there exists at least one jointly fully-colored elementary (n-1)-simplex $\Delta^* = \operatorname{conv}(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*) \in T$ under the coloring functions L_1, \ldots, L_n . Consequently, by exploiting the construction of these coloring functions, for any $i \in [n]$ and for some permutations $\sigma, \tau : [n] \to [n]$ (independent on i), the fractional bundle indexed by $\tau(i)$ is the most valuable bundle for agent i in the fractional partition $\mathcal{S}(m{x}^*_{\sigma(i)})$ derived from the $\sigma(i)$ -th vertex $m{x}^*_{\sigma(i)}$ of Δ^* , under virtual valuation \tilde{v}_i . Starting from the fractional partition $\mathcal{S} := \mathcal{S}(\boldsymbol{x}_1^*)$ associated with the first vertex \boldsymbol{x}_1^* of Δ^* , we use the same rounding procedure as in the equitability case, to obtain a partition $S = (S_1, \ldots, S_n)$ of [m]made of integral connected bundles, referred to as connected integral partition. The claims of Lemma 3.1 and 3.2 obviously apply to \tilde{S} , and can be restated as follows:

Lemma C.1. S is a connected integral partition.

Lemma C.2. For any $h, i, j \in [n]$, the connected (integral) partition S satisfies $v_i^-(S_j) \leq \tilde{v}_i(\tilde{S}_j(\boldsymbol{x}_h^*)) \leq v_i^+(S_j)$.

Now, we can determine the final connected (integral) allocation $\mathcal{A}=(A_1,\ldots,A_n)$ by assigning, for each $i\in[n]$, the bundle $S_{\tau(i)}$ from the connected partition \mathcal{S} to agent i (i.e., $A_i:=S_{\tau(i)}$). We can then prove Theorem 3.5, restated below for completeness:

Theorem C.1 (Statement of Theorem 3.5). Under non-negative valuations, A is an $EF1P_q^c$ allocation.

Proof. First of all, A is a connected allocation by Lemma C.1, as it is obtained by permuting the bundles of S trough permutation τ ($A_i = S_{\tau(i)}$ for any $i \in [n]$). Now, we show the EF1P $_q^c$ guarantee. By the construction of the jointly fully-colored simplex $\Delta^* =$ $\operatorname{conv}(\boldsymbol{x}_1^*,\ldots,\boldsymbol{x}_n^*)$ and the coloring functions L_1,\ldots,L_n , we have that $\tau(i)$ is one of the indices j that maximize the valuation $\tilde{v}_i(S_j(\boldsymbol{x}_{\sigma(i)}^*))$, for any $i \in [n]$ (i.e., the au(i)-th bundle of partition $ilde{\mathcal{S}}(oldsymbol{x}^*_{\sigma(i)})$ maximizes the valuation of agent i among all bundles of that partition). Thus, for any $i, j \in N$, we have $v_i^+(A_i) = v_i^+(S_{\tau(i)}) \ge$ $\tilde{v}_i(\tilde{S}_{\tau(i)}(\boldsymbol{x}^*_{\sigma(i)})) \geq \tilde{v}_i(\tilde{S}_{\tau(j)}(\boldsymbol{x}^*_{\sigma(i)})) \geq v_i^-(S_{\tau(j)}) =$ $v_i^-(A_i)$, where the second inequality follows from the above observation, and the first and last inequalities follow from Lemma 3.2. Since $v_i^+(A_i) \geq v_i^-(A_j)$ for any $i, j \in N$, we conclude that \mathcal{A} satisfies the EF1P $_q^c$ condition, and thus the claim holds.

Remark C.1. As noted in Remark B.1, in the case of equitability one can decide in advance which agent is assigned the *i*-th leftmost bundle. However, unlike in that case, for $EF1P_g^c$ allocations this assignment cannot be predetermined, since the ordering of agents to whom the bundles are assigned from left to right is determined by the jointly fully-colored simplex returned by the Generalized Sperner's Lemma in the proof of Theorem 3.5. This constitutes the main obstacle preventing the extension of the dynamic programming approach used for equitability in Theorem 3.4 to compute $EF1P_g^c$ allocations. Indeed, such an approach would first require knowing the ordering of agents to whom the bundles are assigned from left

to right in the future $EF1P_g^c$ allocation, and then applying a similar computational method as in Theorem 3.4 to obtain such an $EF1P_g^c$ allocation. With this respect, we point out that, differently from the case of equitability, it is not guaranteed that, for any agents ordering to whom the bundles are assigned from left to right, there exists an $EF1P_g^c$ allocation consistent with that assignment.

D Missing Proofs from Section 4

Proof of Theorem 4.1 (Multi-coloring Sperner's Lemma)

We first assume, without loss of generality, that for any vertex \boldsymbol{x} located on the boundary of Δ , the condition $\mathcal{L}(\boldsymbol{x}) = \{i \in [n] : \boldsymbol{x} \in F_i\}$ holds, and that $|\mathcal{L}(\boldsymbol{x})| = 1$ for any internal vertex \boldsymbol{x} not located on the boundary. Indeed, if the claim holds under this restriction, it also holds in the more general case where $\mathcal{L}(\boldsymbol{x}) \supseteq \{i \in [n] : \boldsymbol{x} \in F_i\}$ for boundary vertices \boldsymbol{x} and $|\mathcal{L}(\boldsymbol{x})| \ge 1$ for the other vertices.

Let L be the standard coloring function associated with the multi-coloring one \mathcal{L} , that assigns to each vertex $x \in V(T)$ the color $L(x) = \min\{i \in \mathcal{L}(x)\}$; we refer to L as the *minimal restriction* of \mathcal{L} . We will show, by induction on $n \geq 2$, that the number of fully-colored simplices with respect to the minimal restriction L is odd, which implies the existence of at least one such simplex. By the construction of L, any fully-colored simplex with respect to L is also fully-colored with respect to the original multicoloring function \mathcal{L} , thereby proving the claim of the theorem.

Base Case (n=2): In this case, Δ is the 1-simplex $[v_1,v_2]$, whose 0-dimensional faces are the two endpoint vertices v_1 and v_2 . The triangulation T can be viewed as a path consisting of vertices V(T), connected by edges that correspond to the (contiguous) elementary 1-simplices [x,y] of T. We have $L(x) \in [2]$ for every $x \in V(T)$. Moreover, since $\mathcal L$ is special, and v_1 (resp., v_2) represents the (n-2)-face opposite to v_2 (resp., v_1), it follows that $L(v_1)=2$ (resp., $L(v_2)=1$). Thus, since L assigns colors in $\{1,2\}$ and the colors assigned by L to the endpoints v_1 and v_2 are different, we immediately have that the number of elementary 1-simplices [x,y] with $L(x) \neq L(y)$ (i.e., those which are fully-colored w.r.t. L) must be odd.

Inductive Step: Assume that the claim holds for k=n-1, and let us show it for k=n. Let $F'=\bigcup_{i\in[n-1]}F_i$ be the union of all the (n-2)-faces of Δ , except for the n-th face. The remainder of the proof proceeds with the following two sub-steps:

• Sub-step 1 (Recovering the inductive hypothesis): We have that the triangulation T' induced on F' by T

is topologically equivalent² to the triangulation of an (n-2)-simplex ($\Delta' = \operatorname{conv}(v_1', \dots, v_{n-1}')$). This is established via a homeomprphism f that maps each vertex v_i to v_i' for $i \in [n-1]$, sends the vertex v_n into the interior of Δ' , and maps each (n-3)-face $F_i \cap F_n$ to an (n-3)-face of Δ' . In particular, such a homeomorphism f can be obtained by projecting each point located on F' onto the hyperplane containing the face F_n , parallely to the axis connecting vertex v_n to the barycenter v_n' of F_n . See Figure 5 for a visualization of the projection f in the case n=4.

Thus, we can topologically regard F' as an (n-2)-simplex Δ' and interpret T' as its associated triangulation.

Let $\mathcal{L}': V(T') \to 2^{[n-1]} \setminus \{\emptyset\}$ be the multi-coloring function that assigns to each vertex $x \in V(T')$ the set $\mathcal{L}(x) \setminus \{n\}$, that is, the restriction of \mathcal{L} to the first n-1 colors. By relying on the fact that \mathcal{L} is a special multi-coloring function for T, we have that \mathcal{L}' is also a special multi-coloring function for T', and its minimal restriction L' coincides with the minimal restriction L of \mathcal{L} , when restricted to vertices in V(T') (i.e., located on F'). Thus, we can apply the inductive hypothesis to the triangulation T'and the multi-coloring function L' on V(T'), since $\dim(\Delta') = n - 1$. This allows us to conclude that the number of fully-colored elementary (n-2)simplices in T' with respect to the coloring function L' and using colors from [n-1] is odd. Equivalently, we have that the number of fully-colored elementary (n-2)-simplices in F', with respect to the coloring function L and using colors from [n-1], is odd.

• Sub-step 2 (From dimension n-2 to n-1 through a parity argument on graphs): Let G be the undirected graph whose nodes represent the elementary (n-1)-simplices of Δ , with edges connecting pairs of (n-1)-simplices that share an (n-2)-

²Given two topological spaces X,Y, a homeomorphism $f:X\to Y$ is a bijective continuous function whose inverse is also continuous. Two triangulations T_X and T_Y of topological spaces X and Y, respectively, are topologically equivalent if there exists a homeomorphism $f:X\to Y$ that maps each simplex of T_X to a simplex of T_Y , preserving the adjacency relation, i.e., $f(\Delta_1\cap\Delta_2)=f(\Delta_1)\cap f(\Delta_2)$ for any simplices $\Delta_1,\Delta_2\in T_X$.

dimensional face that is fully-colored with respect to L using colors from [n-1]. By leveraging the construction of the edges, we see that no node in G has a degree greater than 2. This means that G consists of a union of undirected paths or cycles (excluding isolated nodes). Consequently, the number of nodes with degree 1 in G must be even. Moreover, we observe that a node has degree 1 in G if and only if it corresponds to an elementary (n-1)-simplex of one of the following types: (i) an almost fully-colored (n-1)-simplex on the boundary, where each color $i \in [n-1]$ appears exactly once, except for one color that appears twice, and that has a face touching F'; (ii) a fully-colored elementary (n-1)-simplex (where each color $i \in [n]$ appears exactly once). The simplices of type (i) are bijectively associated with the elementary (n-2)-simplices established in Substep 1, and thus they appear in an odd number. Since the total number of nodes of degree 1 in G is even, it follows that the number of nodes of type (ii), i.e., the desired fully-colored (n-1)-simplices, must be odd, and this concludes the proof of the theorem.

The Multi-coloring Sperner's lemma, applied to an instance with n=3, along with part of its proof (in particular, Sub-step 2), is illustrated in Figure 4 of Appendix G.

Full Details on Existence of EQ1 \mathbf{P}_g^c allocations.

To show the existence of an EQ1P^c_g allocation, we employ the same framework as in the case of non-negative valuations, with minor modifications. We use the same triangulation T as in that case, but we equip T with the multicoloring function $\mathcal L$ that assign to each vertex $x \in V(T)$ the set $\mathcal L(x)$ of indices j which maximize $\tilde v_j(\tilde A_j(x))$, where $\tilde v_j$ is the virtual valuation defined as in Section 3.

Differently from the case of non-negative valuations, under non-positive valuations the empty bundle is always the best one for each agent. Thus, given $i \in [n]$, since each vertex \boldsymbol{x} located on the (n-2)-face F_i opposite to \boldsymbol{v}_i corresponds to an allocation having its i-th bundle empty, we have that $i \in \mathcal{L}(\boldsymbol{x})$. Then, \mathcal{L} is a special multicoloring function and, by the Multi-coloring Sperner's lemma (Theorem 4.1), there exists an elementary (n-1)-simplex $\Delta^* = \operatorname{conv}(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*)$ that is fully-colored un-

der the multi-coloring function \mathcal{L} . This means there exists a permutation $\sigma:[n]\to[n]$ such that $i\in\mathcal{L}(\boldsymbol{x}^*_{\sigma(i)})$ for each $i\in[n]$. Let $\tilde{\mathcal{A}}(\boldsymbol{x}^*_1),\ldots,\tilde{\mathcal{A}}(\boldsymbol{x}^*_n)$ be the sequence of n fractional allocations determined from Δ^* as in the case of non-negative valuations (see Section 3). By leveraging the construction of \mathcal{L} and the above sequence of fractional allocations, we have that for each $i\in[n], i$ is one of the indices j that maximize $\tilde{v}_j(\tilde{A}_j(\boldsymbol{x}^*_{\sigma(i)}))$ (that is, the i-th bundle of allocation $\tilde{\mathcal{A}}(\boldsymbol{x}^*_{\sigma(i)})$ maximizes the virtual valuation \tilde{v}_j among all bundles $\tilde{A}_j(\boldsymbol{x}^*_{\sigma(i)})$ of that allocation). Thus, by exploiting the same reasoning as in Theorem 3.3, we have that the integral connected allocation \mathcal{A} obtained by rounding the first fractional allocation $\tilde{\mathcal{A}}(\boldsymbol{x}^*_1)$ as in Section 3, satisfies the desired approximate equitability guarantees.

Theorem D.1. A is an $EQ1P_g^c$ connected allocation, if the agents' valuations are non-positive.

Furthermore, by exploiting the same algorithmic framework as in Theorem 3.4, we also obtain the following computational result.

Theorem D.2. An $EQ1P_g^c$ allocation for instances with non-positive valuations can be found in polynomial time $O(n^2m^4)$.

If valuations are monotone non-increasing, there are chores only. Thus, the $\mathrm{EQ1P}_g^c$ guarantee is equivalent to the stronger equitability-up-to-one-chore-over-paths (EQ1P), where each agent i can remove at most one outer chore c from her own bundle A_i to obtain a valuation at least as large as that achieved by each other agent. Then, we obtain the following corollary of Theorem D.2:

Corollary D.1. An EQ1P allocation for instances with monotone non-increasing valuations always exists and can be found in polynomial time.

This existential and computational result complement the result of Misra et al. [2021], that holds for monotone non-decreasing valuations only.

Finally, analougely to the case of non-negative valuations, we have the following corollary of Theorem D.2 and a similar remark to Remark B.1.

Corollary D.2. Under identical non-positive valuations, $EQ1P_g^c$ (or, equivalently, $EF1P_g^c$) allocations always exist and can be efficiently computed in polynomial time $O(n^2m^3)$.

Remark D.1. Under non-positive valuations, for any fixed ordering $\rho: [n] \to [n]$ of the agents, we can compute in polynomial time another $EQ1P_g^c$ allocation that is well-ordered w.r.t. the new agents order, that is, it assigns the left-most bundle to agent $\rho(1)$, the second left-most bundle to agent $\rho(2)$, and so forth.

Full Details on Existence of $EF1P_g^c$ allocations.

Before showing the existence of EF1P^c_g allocations, we first provide another multi-coloring variant of Sperner's lemma, instantiated for the specific Kuhn's triangulation, but applied to n multi-coloring functions, as in the Generalized Sperner's Lemma (Theorem 3.2). Specifically, let $\Delta = \{ \boldsymbol{x} = (x_1, \dots, x_{n-1}) \in \mathbb{R}^{n-1} : 0 \leq x_1 \leq x_2 \leq \dots \leq x_{n-1} \leq m \}$ be the (n-1)-simplex and T be the related Kuhn's triangulation as defined in Section 3. Furthermore, let $\mathcal{L}_1, \dots, \mathcal{L}_n$ be n special multicoloring functions of V(T), as defined in the previous analysis of the EQ1P^c_g guarantee. An elementary (n-1)-simplex $\Delta^* = \text{conv}(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*) \in T$ is jointly fully-colored under $\mathcal{L}_1, \dots, \mathcal{L}_n$ if there exist two permutations $\sigma, \tau : [n] \to [n]$ such that $\tau(i) \in \mathcal{L}_i(\boldsymbol{x}_{\sigma(i)}^*)$ for any $i \in [n]$.

Theorem D.3 (Generalized Multi-coloring Sperner's Lemma). Let $\mathcal{L}_1, \ldots, \mathcal{L}_n$ be n special multi-coloring functions of the Kuhn's triangulation T of the (n-1)-simplex Δ . Then, there exists a jointly-fully-colored elementary (n-1)-simplex $\Delta^* \in T$ under multi-coloring functions $\mathcal{L}_1, \ldots, \mathcal{L}_n$.

Proof. Given two vertices \boldsymbol{x} and \boldsymbol{y} , a *simple path* P from \boldsymbol{x} to \boldsymbol{y} in T is a sequence of adjacent elementary 1-simplices of T connecting \boldsymbol{x} and \boldsymbol{y} , and its *length* is given by the number of such simplices. Let $\Phi:V(T)\to [n]$ be an auxiliary standard coloring function such that $\Phi(\boldsymbol{x})=1+(d(\boldsymbol{x},\boldsymbol{0}_{n-1}) \bmod n)$, where $d(\boldsymbol{x},\boldsymbol{y})$ is the length of the shortest path connecting \boldsymbol{x} and \boldsymbol{y} , $\boldsymbol{0}_{n-1}=(0,\ldots,0)$ is the origin vertex, and "mod n" denote the remainder

operator modulo n. Let $\overline{\mathcal{L}}$ be a new aggregated multi-coloring function that assigns color $\overline{\mathcal{L}}(x) := \mathcal{L}_{\Phi(x)}(x)$ to each vertex $x \in V(T)$, where $\mathcal{L}_1, \ldots, \mathcal{L}_n$ are the input special multi-coloring functions. We observe that, since each \mathcal{L}_i is special, then $\overline{\mathcal{L}}$ is special, too. Then, by relying on the standard Multi-coloring Sperner's lemma (Theorem 4.1), there exists an elementary (n-1)-simplex $\Delta^* = \operatorname{conv}(x_1^*, \ldots, x_n^*) \in T$ that is fully-colored under $\overline{\mathcal{L}}$, i.e., there exists a permutation $\pi: [n] \to [n]$ such that $j \in \overline{\mathcal{L}}(x_{\pi(j)}^*)$ for any $j \in [n]$.

It remains to show that $\overline{\mathcal{L}}$ refers to a distinct multicoloring function \mathcal{L}_i at each distinct vertex $\boldsymbol{x}_{\sigma(i)}^*$, for some permutation $\sigma:[n]\to[n]$. Together with the fact that Δ^* is fully-colored w.r.t. $\overline{\mathcal{L}}$, this will imply that Δ^* is jointly fully-colored w.r.t. the multi-coloring functions $\mathcal{L}_1,\ldots,\mathcal{L}_n$.

By construction of Φ , we have that Φ assigns a distinct color in [n] to each vertex of Δ^* . Indeed, for any vertices $x, y \in V(T)$, the distance d(x, y) between xand y (used to describe Φ) can be equivalently defined as the scaled Manhattan distance³ $3\sum_{h=1}^{n-1}|x_h-y_h|$. Furthermore, by exploiting the structure of the elementary (n-1)-simplices in the Kuhn's triangulation T and the definition of Manhattan distance, we have that, given an elementary (n-1)-simplex of T, after an appropriate reordering of its vertices, each vertex is at distance 1 from the previous one according to the above scaled Manhattan distance, and the *i*-th-vertex in this ordering is at distance i-1 from the first vertex. Thus, Φ attains distinct values in [n] at each of the n vertices of Δ^* , then allowing to define a permutation $\sigma: [n] \to [n]$ such that $\Phi(x^*_{\sigma(i)}) = i$ for any $i \in [n-1]$.

We recall that $j \in \overline{\mathcal{L}}(\boldsymbol{x}_{\pi(j)}^*) = \mathcal{L}_{\Phi(\boldsymbol{x}_{\pi(j)}^*)}(\boldsymbol{x}_{\pi(j)}^*)$ holds for any $j \in [n]$. Let $\tau := \pi^{-1} \circ \sigma$, where σ is the permutation such that $\Phi(\boldsymbol{x}_{\sigma(i)}^*) = i$ for any $i \in [n-1]$. By using $\tau(i)$ in place of j in the above membership relation involving $\overline{\mathcal{L}}$, we have $\tau(i) \in \mathcal{L}_{\Phi(\boldsymbol{x}_{\pi(\tau(j))}^*)}(\boldsymbol{x}_{\pi(\tau(i))}^*) = \mathcal{L}_{\Phi(\boldsymbol{x}_{\sigma(i)}^*)}(\boldsymbol{x}_{\sigma(i)}^*) = \mathcal{L}_i(\boldsymbol{x}_{\sigma(i)}^*)$ for any $i \in [n]$, where the first equality holds by $\tau = \pi^{-1} \circ \sigma$ and the last one holds by $\Phi(\boldsymbol{x}_{\sigma(i)}^*) = i$. We conclude that Δ^* is jointly fully-colored under $\mathcal{L}_1, \dots, \mathcal{L}_n$.

³Given two vectors $(x_1,\ldots,x_k),(y_1,\ldots,y_k)\in\mathbb{R}^k$, the Manhattan distance between \boldsymbol{x} and \boldsymbol{y} is given by $\sum_{h=1}^k|x_h-y_h|$.

To establish the existence of an EF1P $_g^c$ allocation, we adopt the same framework as in the approximate equitability case, with minor modifications. Specifically, we use the same triangulation T as before but equip it with n distinct multi-coloring functions, $\mathcal{L}_1, \ldots, \mathcal{L}_n$ (defined below), instead of the single multi-coloring function \mathcal{L} used earlier.

As in the case of the approximate envy-freeness guarantee for non-negative valuations, each vertex $x \in V(T)$ now corresponds to a fractional connected partition $\tilde{S}(x) = (\tilde{S}_1(x), \dots, \tilde{S}_n(x))$ of the interval [0, m], where each fractional bundle is initially unassigned to any agent (unlike in the equitability case). For any $i \in [n]$, we consider a distinct multi-coloring function \mathcal{L}_i that assigns to each vertex $x \in V(T)$ the set $\mathcal{L}_i(x)$ of indices j corresponding to the bundles $\tilde{S}_j(x)$ that maximize the valuation $\tilde{v}_i(\tilde{S}_j(x))$ of agent i, where \tilde{v}_i is the virtual valuation considered above. As in the case of equitability for non-positive valuations, each \mathcal{L}_i is again a special multi-coloring function, since the empty bundle is the most valuable for any agent.

Theorem D.3 (Generalized Multi-coloring Sperner's lemma), there exists at least jointly fully-colored elementary (n-1)-simplex $\Delta^* = \operatorname{conv}(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*) \in T$ under the coloring functions $\mathcal{L}_1, \ldots, \mathcal{L}_n$. Consequently, by exploiting the construction of these coloring functions, for some permutations $\sigma, \tau : [n] \rightarrow [n]$ and for any $i \in [n]$, we have that $\tau(i)$ is one of the indices j maximizing $\tilde{v}_i(\tilde{S}_j(\boldsymbol{x}_{\sigma(i)}^*))$ (that is, the fractional bundle indexed by $\tau(i)$ is the most valuable bundle for agent i in allocation $\mathcal{S}(\boldsymbol{x}_{\sigma(i)}^*)$, under virtual valuation \tilde{v}_i). Then, we can apply the same rounding procedure as in the case of non-negative valuations (Section 3) to the partition S to obtain a connected integral allocation $\mathcal{A} = (A_1, \dots, A_n)$ of [m], obtained by assigning, for each $i \in [n]$, the bundle $S_{\tau(i)}$ from the connected partition \mathcal{S} to agent i (i.e., $A_i := S_{\tau(i)}$). Then, we obtain the following theorem, analogous to Theorem 3.5:

Theorem D.4. Under non-positive valuations, A is an $EF1P_q^c$ allocation.

If valuations are monotone non-increasing, the EF1P $_g^c$ guarantee is equivalent to the stronger *envy-freeness-up-to-one-chore-over-paths* (EF1P), where each agent i can

remove at most one outer chore c from her own bundle A_i to obtain a valuation at least as large as that achieved for the other bundles. Then, we obtain the following corollary:

Corollary D.3. An EF1P allocation for instances with monotone non-increasing valuations always exists.

This result complements the findings of Bilò et al. [2022], Igarashi [2023], that hold for monotone non-decreasing valuations only.

E Objective Valuations

We recall that, under objective valuations, any item x is either a good or a chore (independently on the considered agents and bundles); in such a case, we can partition M into a set of goods G and a set of chores C, and we choose arbitrarily how to classify dummy items that qualify as both.

For the remainder of this section, we assume w.l.o.g. that $m \ge n-1$. Indeed, the following remark explains how to handle the case m < n-1.

Remark E.1. If the initial instance I had n-1>m, we could remove n-m-1 arbitrary players, thereby obtaining a new instance I' with n':=m+1>0 players. Let \mathcal{A}' be an allocation in I' that satisfies one of the considered approximate equitability criteria. We observe that the allocation \mathcal{A} obtained from \mathcal{A}' by assigning empty bundles to all n-m-1 excluded players also satisfies the same criterion. Since I' has m items and m+1 players, at least one player i in \mathcal{A}' necessarily receives the empty bundle. Therefore, when extending \mathcal{A}' to \mathcal{A} , the fairness guarantees established in \mathcal{A}' are preserved, as the excluded players also receive empty bundles and can be treated analogously to player i.

In light of this observation, any polynomial or pseudopolynomial algorithm for computing approximately equitable allocations in instances with $m \geq n-1$ can be adapted to instances I with n-1 > m by applying it to the reduced instance I' with n' = O(m) players. Consequently, in this case, the dependence of the running time on n can be replaced by a dependence on m = O(n).

In Algorithm 2 of Appendix H, we present a *local-search algorithm* that returns an EQX $_g^c$ allocation in pseudo-polynomial time under objective valuations; this algorithm is a variant of the local-search framework of Barman et al. [2024]. Algorithm 2 starts from an allocation assigning all items to the first agent. Then, it executes either the while-loop at lines 4–7 if $v_1(M)>0$, or the while-loop at lines 12–15 if $v_1(M)<0$, and simply returns the initial allocation otherwise (i.e., if $v_1(M)=0$). The former while-loop, referred to as *good-moving*, repeatedly moves goods from the bundle of some envied

agent (in terms of inequity) to the least valuable bundle. The latter, referred to as *chore-moving*, repeatedly moves chores from the bundle of some envious agent to the bundle of the most envied one. Both loops terminate when the EQX_a^c condition is satisfied⁵.

Theorem E.1. Given an allocation instance $I = (N, M = G \cup C, (v_i)_{i \in N})$ with objective valuations, the local-search algorithm returns an EQX_g^c allocation in pseudo-polynomial time $O(V_{max} \cdot n \cdot m^2)$, with $V_{max} := \max_{i \in N} \max\{v_i(G), |v_i(C)|\}$.

Sketch of the proof. The full proof of the theorem is deferred to Appendix F; here, we provide the main intuitions only. If $v_1(M) = 0$, the allocation returned by the algorithm is trivially EQ, and then EQX_q^c . If $v_1(M) > 0$ or $v_1(M) < 0$, one can first show that the violation of the while-condition of each while-loop in the local-search algorithm is equivalent to finding an EQX_q^c allocation (Lemma F.3 of Appendix F). By this observation, to prove that the algorithm returns an EQX $_q^c$ allocation, it suffices to show that the executed while-loop (good-moving or chore-moving) terminates. To establish termination, we use a potential function argument. Specifically, we define two ad-hoc potential functions, one for each while-loop. Each potential function depends only on the current allocation, returns a triplet of values, and increases in every iteration (refer to Lemma F.4 and F.5 of Appendix F), according to a lexicographic order ≻ of the triplet values (i.e., $T_1 > T_2$ if and only if the first differing component between triplets $T_1 = (x_1, y_1, z_1)$ and $T_2 = (x_2, y_2, z_2)$ has a higher value in T_1). Since each potential function is

⁴We note that these arguments do not, in general, apply under exact or approximate envy-freeness.

⁵In the full proof, for completeness, we analyze both the cases $v_1(M) > 0$ and $v_1(M) < 0$, corresponding to the execution of the good-moving while-loop and the chore-moving while-loop, respectively. Anyway, to handle the case $v_1(M) < 0$, one could directly appeal to a mirroring argument with respect to the case $v_1(M) > 0$. Indeed, consider a fair allocation instance with valuations v_i' obtained by multiplying the original valuations v_i by -1. We observe that: the case $v_1'(M) > 0$ corresponds to $v_1(M) < 0$; executing the good-moving while-loop with respect to v_i' is equivalent to executing the chore-moving while-loop with respect to the original valuations v_i (since inequalities reverse, indices i,j are exchanged, and goods become chores); and property (i) of EQX_g^c under v_i' is equivalent to property (ii) of EQX_q under v_i . Thus, when $v_1(M) < 0$, the ability of the goodmoving while-loop to compute allocations satisfying property (i) of the EQX_{a}^{c} guarantee under valuations v_{i}^{\prime} can be directly translated into the ability of the chore-moving while-loop to compute allocations satisfying property (ii) of the EQX $_q^c$ guarantee under the original valuations v_i .

necessarily bounded (e.g., by the maximal triplet achievable across all possible allocations), it follows that each while-loop must terminate. For the good-moving while-loop (i.e., case $v_1(M)>0$), we consider the potential function Φ that assigns, to each allocation \mathcal{A} , the triplet $(x(\mathcal{A}),y(\mathcal{A}),z(\mathcal{A}))$, such that $x(\mathcal{A})$ is the minimum valuation in \mathcal{A} , $y(\mathcal{A})$ is minus the number of agents obtaining the minimum valuation, and $z(\mathcal{A})$ is the number of items allocated to such agents. For the chore-moving while loop (i.e., case $v_1(M)<0$), we consider the potential function Ξ that assigns, to each allocation \mathcal{A} , the triplet $(x'(\mathcal{A}),y'(\mathcal{A}),z'(\mathcal{A}))$, such that $x'(\mathcal{A})$ is minus the maximum valuation in \mathcal{A} , $y'(\mathcal{A})$ is minus the number of agents obtaining the maximum valuation, and $z'(\mathcal{A})$ is the number of items allocated to such agents.

To estimate the time complexity of the algorithm, we observe that it is equal to the time required for each iteration of the while-loops, multiplied by the total number of iterations. The while-condition can be checked in time O(m+n)=O(m), as it requires to check all bundles, and all items for each bundle⁶.

To bound the number of iterations, we first observe that the potential function increases at each iteration. Consequently, the number of iterations is bounded by the maximum value of the potential functions Φ and Ξ , that is $O(V_{\max} \cdot n \cdot m)$ (as established in Lemma F.6 in Appendix F).

Thus, the overall time complexity is $O(V_{\max} \cdot n \cdot m^2)$

By observing that, under monotone non-decreasing (resp. non-increasing) valuations, property (i) (resp. (ii)) of the EQX $_g^c$ guarantee is equivalent to the stronger EQX notion, we obtain the following corollary of Theorem E.1.

Corollary E.1. The local-search algorithm returns an EQX allocation in pseudo-polynomial time, under monotone (non-decreasing and non-increasing) valuations.

In the following, we present a *greedy algorithm* that returns in polynomial time an EQ1 allocation under objective valuations (for the pseudo-code, see Algorithm 3 in Appendix F). This algorithm can be seen as a straightforward generalization of the greedy approach introduced

by Hosseini and Sethia [2025], which was specifically designed for the case of objective additive valuations.

Starting from the empty allocation, the algorithm first processes all goods, and repeatedly assigns each good (in an arbitrary order) to the least valuable bundle of the current iteration. Then, it processes all chores, and repeatedly assigns each chore to the most valuable bundle of the current iteration. We observe that the initial local-search algorithm clearly returns an EQ1 allocation but operates in pseudo-polynomial time, whereas the greedy algorithm typically runs faster, operating in polynomial time.

Theorem E.2. Given an allocation instance $I = (N, M = G \cup C, (v_i)_{i \in N})$ with objective valuations, the greedy algorithm returns an EQ1 allocation in polynomial time $O(m \log n)$.

Sketch of the proof. It can be shown by induction that the partial allocation \mathcal{A}^t obtained after assigning the first t items is EQ1, and consequently, the final allocation $\mathcal{A} = \mathcal{A}^t$ is also EQ1. Full details are deferred to Appendix F.

Regarding the time complexity, the algorithm performs at most O(m) item insertions. For each insertion, it identifies and updates either the least or the most valuable bundle among n, which can be done in $O(\log n)$ time using a heap-tree data structure. Therefore, the overall time complexity is $O(m \log n)$.

Finally, we consider a variant of the above greedy algorithm, called strongly-greedy algorithm, that is specialized for objective valuations that are also additive (for the pseudo-code, refer to Algorithm 4 in Appendix F). This algorithm operates like the standard greedy algorithm, but in each iteration, when a good (or a chore) is assigned to an agent, it selects the one that maximizes (or minimizes) that agent's valuation. Under objective additive valuations, the strongly-greedy algorithm returns an EQX_g^c allocation, ensuring stronger equity than the EQ1 guarantee provided by Hosseini and Sethia [2025] or the standard greedy algorithm (Algorithm 3 in Appendix H).

Theorem E.3. Given an allocation instance $I = (N, M = G \cup C, (v_i)_{i \in N})$ with additive objective valuations, the strongly-greedy algorithm returns an EQX_g^c allocation in polynomial time $O(mn \log m)$.

Theorem E.3 can be shown in a manner similar to Theorem E.2, with a sketch of the proof provided in Appendix F.

 $^{^6}O(m+n)=O(m)$ holds since we assumed w.l.o.g. that n=O(m) (by Remark E.1 of Appendix E).

F Missing Proofs from Appendix E

Full Proof of Theorem E.1 - Correctness

We first show the correctness of the algorithm. By relying on the two technical lemmas provided below (Lemmas F.1 and F.2), we will first show that the violation of the while-condition of each while-loop is equivalent to finding an EQX_a^c allocation (Lemma F.3).

Lemma F.1. Under objective valuations, for any $i \in N$ and non-empty bundle $S \subseteq M$ such that S contains goods (resp. chores) only, we have $v_i(S) \ge 0$ (resp. $v_i(S) \le 0$). Equivalently, if i and S satisfy $v_i(S) > 0$ (resp. $v_i(S) < 0$), then there exists at least one good (resp. one chore) in S.

The proof of Lemma F.1 is straightforward and directly follows from the definition of goods and chores.

Lemma F.2. Assume that $v_1(M) > 0$ (resp. $v_1(M) < 0$), and let $A^t = (A_1^t, \dots, A_n^t)$ denote the allocation obtained after each iteration $t \ge 1$ of the good-moving (resp. chore-moving) while-loop, with A^0 denoting the initial allocation. Then, for any $t \ge 0$, we have that A_1^t contains at least one good (resp. chore) and each non-empty bundle A_b^t with $h \ne 1$ contains only goods (resp. chores).

Proof of Lemma F.2. We first show the claim in the case $v_1(M)>0$, and we proceed by induction on $t\geq 0$. We start with t=0 (base case). We have $v_1(A_1^0)=v_1(M)>0$, since $A_1^0=M$ and we are in the case $v_1(M)>0$. Thus, by Lemma F.1, we have that A_1^0 contains at least one good. Since $A_h^0=\emptyset$ for any $h\neq 1$, the above property is sufficient to establish the claim for t=0.

Now, assume that the claim holds for an arbitrary t (inductive hypothesis), and let us show it for t+1 (inductive step). For all agents h such that $A_h^t = A_h^{t+1}$ (i.e., whose bundle remains unchanged in iteration t+1) the claim trivially holds by the inductive hypothesis. Next, we focus on agents who either receive or lose one good in each iteration t+1. Let h be any such an agent. If $h \neq 1$, by the inductive hypothesis, we have that A_h^t contains goods only. So, if h receives a good, then A_h^{t+1} , if not empty, also contains goods only. Now, assume that h=1. By the inductive hypothesis, we have that A_1^t contains at least one good. If agent 1 receives a good in iteration t+1, we then have that A_1^{t+1} contains at least one

good as well. If agent 1 loses a good in iteration t+1, by lines 4 and 5 of Algorithm 2, there exist an agent i and a good $g \in A_1^t$ such that $v_1(A_1^t \setminus \{g\}) > v_i(A_i^t)$. We derive $v_1(A_1^{t+1}) = v_1(A_1^t \setminus \{g\}) > v_i(A_i^t) \geq 0$, where the last inequality holds since A_i^t contains goods only (by Lemma F.1 and the inductive hypothesis). Then, since $v_1(A_1^{t+1}) > 0$, Lemma F.1 implies that A_1^{t+1} contains at least one good. We conclude that the inductive step holds for any agent h, and the claim follows in the case $v_1(M) \geq 0$.

For the case $v_1(M)<0$, the claim can be shown by using an argument symmetric to that of the previous case. Let v_i' be the valuation obtained by multiplying the value of v_i by minus one, that is, $v_i'(S)=-v_i(S)$ for any $i\in N$ and $S\subseteq M$. We observe that $v_1'(M)=-v_1(M)>0$. Thus, under the valuations v_i' , the previous analysis implies that, for any $t\geq 0$, the bundle A_h^t contains at least one good, while each non-empty bundle A_h^t with $h\neq 1$ contains only goods. Returning to the original valuations v_i , this implies that, for any $t\geq 0$, A_1^t contains at least one chore, while each non-empty bundle A_h^t with $h\neq 1$ contains only chores. \square

Lemma F.3. Assume that $v_1(M) > 0$ (resp. $v_1(M) < 0$). If A is an allocation found by the good-moving (resp. chore-moving) while-loop that violates the corresponding while-condition, then A is EQX_q^c .

Proof of Lemma F.3. We first assume that $v_1(M) > 0$. We will show that, if the while-condition is violated, property (i) of the EQX_g^c guarantee is satisfied for any $i \in N$ (that is, the inequity an agent i feels toward the bundle assigned to another agent j can be eliminated by removing any good from j's bundle). To establish this, it suffices to show that every non-empty bundle in $\mathcal{A} = (A_1, \dots, A_n)$ contains at least one good, as the remaining part of property (i) of the EQX_g^c guarantee is ensured by the violation of the while-condition. This fact is immediately guaranteed by Lemma F.2, and thus the claim follows.

To show the claim when $v_1(M) < 0$, we use an argument symmetric to the previous case. Specifically, we will show that, if the while-condition is violated, property (ii) of the EQX $_g^c$ guarantee is satisfied for any $i \in N$ (that is, the inequity an agent i feels toward the bundle assigned to another agent j can be eliminated by removing any chore from i's bundle). To establish this, it suffices to

show that every non-empty bundle in $\mathcal{A}=(A_1,\ldots,A_n)$ contains at least one chore, as the remaining part of property (ii) of the EQX $_g^c$ guarantee is ensured by the violation of the while-condition. Again, this fact is guaranteed by Lemma F.2, thus proving the claim.

We now move to the main part of the proof. If $v_1(M) =$ 0, the allocation returned by the algorithm assigns all items to the first agent. Such an allocation is trivially EQ, and therefore EQX_q^c , as each agent receives zero value. Thus, in the remainder of the proof we focus on the cases $v_1(M) > 0$ and $v_1(M) < 0$. By Lemma F.3, to prove that Algorithm 2 returns an EQX_g^c allocation in these cases, it suffices to show that the executed while-loop (goodmoving or chore-moving) terminates. To establish termination, we use a potential function argument. Specifically, we define two ad-hoc potential functions, one for each while-loop. Each potential function depends only on the current allocation, returns a triplet of values, and increases in every iteration, following a lexicographic order of the triplet values. Since each potential function is necessarily bounded (e.g., by the maximal triplet achievable across all possible allocations), it follows that the while-loop must terminate. We treat separately the cases $v_1(M) > 0$ and $v_1(M) < 0$, each corresponding to a distinct potential function, though the proof arguments are analogous.

Case $v_1(M) > 0$: We first consider the case $v_1(M) > 0$ 0. Let Φ be the potential function that assigns, to each allocation \mathcal{A} , the triplet $(x(\mathcal{A}), y(\mathcal{A}), z(\mathcal{A}))$, where $x(\mathcal{A}) =$ $\min_{i \in N} v_i(A_i), y(A) = -|\{i \in N : v_i(A_i) = x(A)\}|$ and $z(A) = \sum_{i \in N: v_i(A_i) = x(A)} |A_i|$. Specifically, x(A)is the minimum valuation in A, -y(A) is the number of agents obtaining the minimum valuation, and z(A) is the number of items allocated to such agents. We consider the total order over triplets such that $(x_1, y_1, z_1) \succ$ (x_2, y_2, z_2) iff (x_1, y_1, z_1) is lexicographically higher than (x_2, y_2, z_2) , i.e., if $x_1 > x_2 \lor (x_1 = x_2 \land y_1 > x_2)$ $(y_2) \lor (x_1 = x_2 \land y_1 = y_2 \land z_1 > z_2)$. Assume by contradiction that the good-moving while-loop never terminates. Let A^t denote the allocation computed after each iteration $t \geq 0$, where \mathcal{A}^0 denotes the initial allocation, and t = 0denotes the state prior to the first iteration. We have the following lemma:

Lemma F.4. Assume that $v_1(M) > 0$ and let $t \geq 0$ be an iteration of the good-moving while-loop such that the allocation A_t obtained at the end of the iteration satisfies again the while-condition. Then, we have $\Phi(A^{t+1}) \succ \Phi(A^t)$.

Proof of Lemma F.4. Let $t \geq 0$ be the iteration satisfying the hypothesis of the lemma. Then, there exist two agents $i,j \in N$ and a good $g \in A_j$ such that i has the least valuable bundle in \mathcal{A}^t , $v_j(A_j^t \setminus \{g\}) > v_i(A_i^t)$, \mathcal{A}^{t+1} is obtained from \mathcal{A}^t by moving good g from the bundle of j to the bundle of i, and the other bundles do not vary. One of the following three cases can occur:

- (i) $v_i(A_i^{t+1}) = v_i(A_i^t)$: By $v_j(A_j^{t+1}) = v_j(A_j^t \setminus \{g\}) > v_i(A_i^t) = v_i(A_i^{t+1})$, the bundle of i (resp. j) is (resp. is not) the least valuable bundle in both \mathcal{A}^t and \mathcal{A}^{t+1} . Thus, the value of the least valuable bundle does not vary, i.e., $x(\mathcal{A}^{t+1}) = x(\mathcal{A}^t)$, the number of least valuable bundles remains the same, i.e., $-y(\mathcal{A}^{t+1}) = -y(\mathcal{A}^t)$, but the overall cardinality of the least valuable bundles increases by one, i.e., $z(\mathcal{A}^{t+1}) = z(\mathcal{A}^t) + 1 > z(\mathcal{A}^t)$ (since $|A_i^{t+1}| = 1 + |A_i^t|$, and the other eventual least valuable bundles do not change). We conclude that $\Phi(\mathcal{A}^{t+1}) \succ \Phi(\mathcal{A}^t)$ holds in case (i).
- (ii) $v_i(A_i^{t+1}) > v_i(A_i^t)$ and $-y(\mathcal{A}^t) > 1$: Given an arbitrary agent $h \neq i$ possessing one of the least valuable bundles in \mathcal{A}^t , we have $v_i(A_i^t) = v_h(A_h^t) =$ $v_h(A_h^{t+1})$, where the last equality holds since the bundle of h does not vary in iteration t + 1; furthermore, such an agent $h \neq i$ exists, since the number of least valuable bundles in \mathcal{A}^t is $-y(\mathcal{A}) > 1$. Then, by $v_j(A_j^{t+1}) = v_j(A_j^t \setminus \{g\}) > v_i(A_i^t) = v_h(A_h^{t+1})$ and $v_i(A_i^{t+1}) > v_i(A_i^t) = v_h(A_h^{t+1})$, the following facts hold: the bundle of j is not the least valuable bundle in both A^t and A^{t+1} ; the bundle of i is the least valuable bundle in A^t but not in A^{t+1} ; the bundle of h is the least valuable bundle in both \mathcal{A}^t and \mathcal{A}^{t+1} . By the arbitrariness of the choice of agent $h \neq i$ possessing one of the least valuable bundles in A^t , we conclude that the minimum value $x(\mathcal{A}^{t+1})$ among all bundles in \mathcal{A}^{t+1} continues to be equal to $x(\mathcal{A}^t) = v_i(A_i^t)$, and is achieved by all agents h obtaining such value in A^t but i, i.e., $-y(\mathcal{A}^{t+1}) = -y(\mathcal{A}^t) - 1 \ge 1$. We conclude that

 $x(\mathcal{A}^{t+1}) = x(\mathcal{A}^t) \text{ and } y(\mathcal{A}^{t+1}) = y(\mathcal{A}^t) + 1 > y(\mathcal{A}^t), \text{ that is, } \Phi(\mathcal{A}^{t+1}) \succ \Phi(\mathcal{A}^t) \text{ holds in case (ii).}$

• (iii) $v_i(A_i^{t+1}) > v_i(A_i^t)$ and $-y(\mathcal{A}^t) = 1$: Since the number of least valuable bundles in \mathcal{A}^t is $-y(\mathcal{A}^t) = 1$, we have that A_i^t is the unique bundle minimizing the valuation in \mathcal{A}^t . Then, by $v_i(A_i^{t+1}) > v_i(A_i^t)$ and $v_j(A_j^{t+1}) = v_j(A_j^t \setminus \{g\}) > v_i(A_i^t)$, we necessarily have that that the minimum valuation $x(\mathcal{A}^{t+1})$ in \mathcal{A}^{t+1} is strictly higher than the minimum valuation $x(\mathcal{A}^t) = v_i(A_i^t)$ in \mathcal{A}^t , that is, $\Phi(\mathcal{A}^{t+1}) \succ \Phi(\mathcal{A}^t)$ holds in case (iii).

In any case, we have $\Phi(\mathcal{A}^{t+1}) \succ \Phi(\mathcal{A}^t)$, thus showing the claim. \square

Under the hypothesis of non-termination of the good-moving while-loop, there exists at least one allocation that is visited twice. In other words, there are at least two iterations $t,t'\geq 0$ with t< t' such that $\mathcal{A}^t=\mathcal{A}^{t'}$, and hence $\Phi(\mathcal{A}^t)=\Phi(\mathcal{A}^{t'})$. Thus, by this observation and Lemma F.4, we have $\Phi(\mathcal{A}^t)=\Phi(\mathcal{A}^{t'})\succ\Phi(\mathcal{A}^{t'-1})\succ\ldots\succ\Phi(\mathcal{A}^{t+1})\succ\Phi(\mathcal{A}^t)$, that leads to the contradiction $\Phi(\mathcal{A}^t)\succ\Phi(\mathcal{A}^t)$. We conclude that the good-moving while-loop necessarily terminates, and then, by Lemma F.3, it returns an EQX $_q^c$ allocation if $v_1(M)>0$.

Case $v_1(M) < 0$: Now, let us consider the case $v_1(M) < 0$. Let Ξ be the potential function that assigns, to each allocation A, the triplet (x'(A), y'(A), z'(A)), where $x'(A) = -\max_{i \in N} v_i(A_i), y'(A)$ $-|\{i \in N : v_i(A_i) = -x'(A)\}|$ and z'(A) = $\sum_{i \in N: v_i(A_i) = -x'(\mathcal{A})} |A_i|$. Specifically, $-x'(\mathcal{A})$ is the maximum valuation in A, -y'(A) is the number of agents obtaining the maximum valuation, and z'(A)is the number of items allocated to such agents. Let > be the lexicographic order among triplets defined above. Assume by contradiction that the chore-moving while-loop never terminates. Let A^t denote the allocation computed after each iteration t > 0, where A^0 denotes the initial allocation, and t = 0 denotes the state prior to the first iteration. We have the following lemma (the proof has an argument symmetric to that of Lemma F.4):

Lemma F.5. Assume that $v_1(M) < 0$ and let $t \ge 0$ be an iteration of the chore-moving while-loop such that the

allocation A_t obtained at the end of the iteration satisfies again the while-condition. Then, we have $\Xi(A^{t+1}) \succ \Xi(A^t)$.

Proof of Lemma F.5. Let $t \geq 0$ be the iteration satisfying the hypothesis of the lemma. Then, there exist two agents $i,j \in N$ and a chore $c \in A_i$ such that j has the most valuable bundle in \mathcal{A}^t , $v_i(A_i^t \setminus \{c\}) < v_j(A_j^t)$, \mathcal{A}^{t+1} is obtained from \mathcal{A}^t by moving chore c from the bundle of i to the bundle of j, and the other bundles do not vary. One of the following three cases can occur:

- (i) $v_j(A_j^{t+1}) = v_j(A_j^t)$: By $v_i(A_i^{t+1}) = v_i(A_i^t \setminus \{c\}) < v_j(A_j^t) = v_j(A_j^{t+1})$, the bundle of j (resp. i) is (resp. is not) the most valuable bundle in both \mathcal{A}^t and \mathcal{A}^{t+1} . Thus, the value of the most valuable bundle does not vary, i.e., $x'(\mathcal{A}^{t+1}) = x'(\mathcal{A}^t)$, the number of most valuable bundles remains the same, i.e., $-y'(\mathcal{A}^{t+1}) = -y'(\mathcal{A}^t)$, but the overall cardinality of the most valuable bundles increases by one, i.e., $z'(\mathcal{A}^{t+1}) = z'(\mathcal{A}^t) + 1 > z'(\mathcal{A}^t)$ (since $|A_j^{t+1}| = |A_j^t| + 1$, and the other eventual most valuable bundles do not vary). We conclude that $\Xi(\mathcal{A}^{t+1}) \succ \Xi(\mathcal{A}^t)$ holds in case (i).
- (ii) $v_j(A_i^{t+1}) < v_j(A_i^t)$ and $-y'(A^t) > 1$: Given an arbitrary agent $h \neq j$ possessing one of the most valuable bundles in \mathcal{A}^t , we have $v_i(A_i^t) =$ $v_h(A_h^t) = v_h(A_h^{t+1})$, where the last equality holds since the bundle of h does not vary in iteration t+1; furthermore, such an agent $h \neq j$ exists, since the number of most valuable bundles in A^t is -y'(A) >1. Then, by $v_i(A_i^{t+1}) = v_i(A_i^t \setminus \{c\}) < v_j(A_j^t) = v_h(A_h^{t+1})$ and $v_j(A_j^{t+1}) < v_j(A_j^t) = v_h(A_h^{t+1})$, the following facts hold: the bundle of i is not the most valuable bundle in both A^t and A^{t+1} ; the bundle of j is the most valuable bundle in A^t but not in A^{t+1} ; the bundle of h is the most valuable bundle in both \mathcal{A}^t and \mathcal{A}^{t+1} . By the arbitrariness of the choice of agent $h \neq j$ possessing one of the most valuable bundles in A^t , we conclude that the maximum value $-x'(A^{t+1})$ among all bundles in A^{t+1} continues to be equal to $-x'(A^t) = v_j(A_j^t)$, and is achieved by all agents h obtaining such value in A^t but j, i.e., $-y'(A^{t+1}) = -y'(A^t) - 1 \ge 1$. We conclude that $x'(A^{t+1}) = x'(A^t)$ and $y'(A^{t+1}) = y'(A^t) + 1 > 0$ $y'(A^t)$, that is, $\Xi(A^{t+1}) \succ \Xi(A^t)$ holds in case (ii).

• (iii) $v_j(A_j^{t+1}) < v_j(A_j^t)$ and $-y'(\mathcal{A}^t) = 1$: Since the number of most valuable bundles in \mathcal{A}^t is $-y'(\mathcal{A}^t) = 1$, we have that A_j^t is the unique bundle maximizing the valuation in \mathcal{A}^t . Then, by $v_j(A_j^{t+1}) < v_j(A_j^t)$ and $v_i(A_i^{t+1}) = v_i(A_i^t \setminus \{c\}) < v_j(A_j^t)$, we necessarily have that that the maximum valuation $-x'(\mathcal{A}^{t+1})$ in \mathcal{A}^{t+1} is strictly lower than the maximum valuation $-x'(\mathcal{A}^t) = v_j(A_j^t)$ in \mathcal{A}^t , that is, $\Xi(\mathcal{A}^{t+1}) \succ \Xi(\mathcal{A}^t)$ holds in case (iii).

In any case, we have $\Xi(\mathcal{A}^{t+1}) \succ \Xi(\mathcal{A}^t)$, thus showing the claim

Under the hypothesis of non-termination of the choremoving while-loop, there are two iterations $t,t'\geq 0$ with t< t' such that $\mathcal{A}^t=\mathcal{A}^{t'}$, that is, $\Xi(\mathcal{A}^t)=\Xi(\mathcal{A}^{t'})$. Thus, as in the previous case, we have $\Xi(\mathcal{A}^t)=\Xi(\mathcal{A}^{t'})\succ\Xi(\mathcal{A}^{t'-1})\succ\ldots\succ\Xi(\mathcal{A}^{t+1})\succ\Xi(\mathcal{A}^t)$, that leads to the contradiction $\Xi(\mathcal{A}^t)\succ\Xi(\mathcal{A}^t)$. We conclude that the chore-moving while-loop necessarily terminates, and then, by Lemma F.3, it returns an EQX $_g^c$ allocation in case $v_1(M)<0$, too.

Full Proof of Theorem E.1 - Complexity

The complexity of Algorithm 2 is proportional to the maximum number of iterations performed by the while-loops, multiplied by the number of steps required to check the while-condition.

The while-condition can be checked in time O(m+n)=O(m), as it requires to check all bundles, and all items for each bundle.

The number of iterations of the while-loops is bounded by the maximum number of increases that the potential functions in Lemmas F.4 and F.5 can undergo, which is $O(V_{max} \cdot n \cdot m)$, by the following lemma:

Lemma F.6. The number of iterations of each while-loop (good-moving and chore-moving) is at most $O(V_{max} \cdot n \cdot m)$.

Proof of Lemma F.6. by Lemma F.4, the potential function Φ lexicographically increases after each iteration. Thus, as Φ returns a triplet in the set $[0, \max_{i \in N} v_i(G)] \times \{-n, \ldots, -1\} \times [m]$, we have that after at most $\max_{i \in N} v_i(G) \cdot n \cdot m \leq V_{max} \cdot n \cdot m$ iterations the

good-moving while loop necessarily terminates. Regarding the chore-moving while-loop, Lemma F.5 implies that the potential function Ξ lexicographically increases after each iteration. Again, as Ξ returns a triplet in the set $[0, \max_{i \in N} |v_i(C)|] \times \{-n, \ldots, -1\} \times [m]$, we have that after at most $\max_{i \in N} |v_i(C)| \cdot n \cdot m \leq V_{max} \cdot n \cdot m$ iterations the chore-moving while loop necessarily terminates. \square

We conclude that the complexity of Algorithm 2 is $O(V_{max} \cdot n \cdot m^2)$.

Full Proof of Theorem E.2

We first focus on the correctness of the algorithm. Let \mathcal{A}^t denote the partial allocation obtained after assigning t items. By exploiting the greedy choice of the algorithm, it can be shown by induction on $t \geq 0$ that each partial allocation \mathcal{A}^t is EQ1.

For t = 0 the claim trivially holds. Assume that the claim holds for iteration t, and let us show it for t+1(inductive step). To prove the EQ1 guarantee, it is sufficient showing that, given $i, j \in N$ such that $v_i(A_i^{t+1}) <$ step t+1 is a good, denoted by g, it is assigned to the agent h having the lowest valuation in \mathcal{A}^t . If h=j, we have that $v_i(A_i^{t+1})=v_i(A_i^t)\geq v_h(A_h^t)=v_j(A_j^t)=$ $v_i(A_i^{t+1} \setminus \{g\})$, where the inequality holds by the minimality of $v_h(A_h^t)$. If $h \neq j$, the bundle assigned to j remains unchanged after iteration t+1 and the bundle assigned to i either remains unchanged (case $h \notin \{i, j\}$) or receives an additional good g (case h = i). By the inductive hypothesis, there exists $x \in A_i^t \cup A_i^t$ such that $v_i(A_i^t \setminus \{x\}) \ge v_j(A_i^t \setminus \{x\})$. Thus, by the previous observations, we have $v_i(A_i^{t+1} \setminus x) \ge v_i(A_i^{t+1} \setminus \{x,g\}) = v_i(A_i^t \setminus \{x\}) \ge v_j(A_j^t \setminus x) = v_j(A_j^{t+1} \setminus x)$, and therefore the EQ1 guarantee continues to hold in A^{t+1} .

Now, assume that the item assigned in iteration t+1 is a chore, denoted by c. In such a case, c is assigned to the agent h having the highest value in the current iteration. If h=i, we have that $v_i(A_i^{t+1}\setminus\{c\})=v_i(A_i^t)=v_h(A_h^t)\geq v_j(A_j^t)=v_j(A_j^{t+1})$, where the inequality holds by the maximality of $v_h(A_h^t)$. If $h\neq i$, the bundle assigned to

i remains unchanged after iteration t+1 and the bundle assigned to j either remains unchanged (case $h\notin\{i,j\}$) or receives an additional chore c (case h=j). By the inductive hypothesis, there exists $x\in A_i^t\cup A_j^t$ such that $v_i(A_i^t\setminus\{x\})\geq v_j(A_j^t\setminus\{x\})$. Thus, by the previous observations, we have $v_i(A_i^{t+1}\setminus\{x\})=v_i(A_i^t\setminus\{x\})\geq v_j(A_j^{t+1}\setminus\{x\})=v_j(A_j^{t+1}\setminus\{x\})$.

In any case, \mathcal{A}^{t+1} is EQ1, and this shows the inductive step. We conclude that each allocation \mathcal{A}^t , and a fortiori the final allocation $\mathcal{A} = \mathcal{A}^m$ obtained after all items have been assigned, satisfies EQ1.

Regarding the time complexity, we observe that the algorithm performs at most O(m) item insertions. During each insertion, it identifies and updates either the least or the most valuable bundle among n. This operation can be executed in $O(\log n)$ time using a heap-tree data structure, that can be preliminary constructed in time O(n). Thus, the overall running time is $O(m \log n + n) = O(m \log n)^7$.

Proof of Theorem E.3

We first show the correctness of the algorithm. Let \mathcal{A}^t denote the partial allocation obtained after the assignment of t items. Mimicking the proof of Theorem E.2, one can easily show by induction on $t \geq 0$ that \mathcal{A}^t satisfies the following approximate equity guarantee: for any $i, j \in N$ such that $v_i(A_i^t) < v_j(A_i^t)$, we have $v_i(A_i^t \setminus \{x\}) \ge$ $v_j(A_i^t \setminus \{x\})$, where x is the last chore assigned to A_i^t if A_i^t contains at least one chore, and x is the last good assigned to A_i^t otherwise. By the greedy selection of each agent and the imposed ordering of items, if bundle A_i^t (resp. A_i^t) contains at least one good (resp. chore), the last good (resp. chore) assigned to the bundle is the least (resp. most) valuable good (resp. chore) within the bundle. This fact, combined with above approximate equity guarantee, implies the following: for any $i, j \in N$ such that $v_i(A_i^t) < v_i(A_i^t)$, we have $v_i(A_i^t \setminus \{x\}) \geq v_i(A_i^t)$ for any chore $x \in A_i^t$ if A_i^t contains at least one chore, and $v_i(A_i^t) \geq v_j(A_i^t \setminus \{x\})$ for any good $x \in A_i^t$ otherwise. This final approximate equity guarantee ensures that every allocation \mathcal{A}^t is EQX_g^c , and, in particular, that the final allocation $\mathcal{A} := \mathcal{A}^m$ also satisfies the EQX $_a^c$ guarantee.

Regarding the complexity, the algorithm can be implemented by creating two heap-tree data structures that keep track of the agent with the minimum and maximum valuation (this can be done in time O(n)), and, for each agent $i \in [n]$, two heap-tree data structures that keep track of the best good and the worst chore (this can be done in time O(mn)). Excluding the above initialization, the execution time of the algorithm is proportional to the number of iterations, i.e., m, multiplied by the complexity of each iteration. To execute each iteration, the algorithm must find the agent i having the minimum or maximum valuation, and determine the best available good or the worst available chore for i. Using the aforementioned heap-trees, these operations can be executed in time $O(\log n + n \log m)$ per iteration: $O(\log n)$ to find the agent maximizing or minimizing the valuation, $O(\log m)$ to extract the best good or worst chore and $O(n \log m)$ to update the n heap-trees associated with the agents (in particular, to ensure that items already extracted in other heaps are treated as unusable in each heap). Therefore, the overall running time is $O(n + mn + m(\log n + n \log m)) = O(mn \log m)$.

 $^{^7}O(m\log n + n) = O(m\log n)$ holds since we assumed w.l.o.g. that $m \ge n-1$.

G Missing Figures

Figure 1

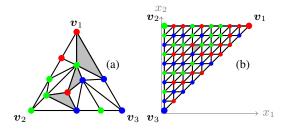


Figure 1: The figure on the left (a) illustrates an application of Sperner's Lemma to a triangulation T of a 2-simplex $\Delta = \operatorname{conv}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$. The coloring function assigns a color from set $\{1 \text{ (Red)}, 2 \text{ (Green)}, 3 \text{ (Blue)}\}$ to each vertex $\boldsymbol{x} \in V(T)$, and it is special. Specifically, for each $i \in [3]$, the color i does not appear on the 1-face of Δ that does not contain \boldsymbol{v}_i (i.e., on the edge opposite to vertex \boldsymbol{v}_i). Sperner's Lemma guarantees the existence of an odd number of fully-colored simplices. In this case, there are three such simplices (that is, there exists at least one such simplex), and they are highlighted in gray. On the right (b), we show Kuhn's triangulation of the 2-simplex $\{(x_1,x_2):0\leq x_1\leq x_2\leq 3\}$, which is based on an allocation instance with n=3 agents and m=3 items. The colors are assigned to vertices based on the special coloring L derived from an arbitrary non-negative virtual valuation functions. One of the fully-colored elementary (n-1)-simplices in T, whose existence is guaranteed by Sperner's Lemma, is highlighted in gray.

Figure 2

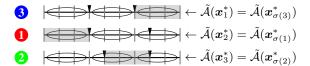


Figure 2: The figure represents the fractional allocations $\tilde{\mathcal{A}}(\boldsymbol{x}_1^*), \tilde{\mathcal{A}}(\boldsymbol{x}_2^*), \tilde{\mathcal{A}}(\boldsymbol{x}_3^*)$ associated with the vertices $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \boldsymbol{x}_3^*$ of the fully-colored simplex Δ^* in Figure 1(b) (highlighted in gray), which are colored blue (label 3), red (label 1), and green (label 2) according to the coloring function L; the permutation $\sigma:[3] \to [3]$ such that $L(\boldsymbol{x}_{\sigma(i)})=i$ is defined by $(1,2,3) \xrightarrow{\sigma} (2,3,1).$ The integral items are represented by ellipses, the fractionality levels of each item (integral, 1-fractional, 2-fractional) are determined by the gray vertical lines, that cut each item in three parts, and the knives that determine the fractional allocations are represented by the black triangles. In such example, as indicated by the vertex colors, the virtual valuation determining L is maximized by the third (resp. first, resp. second) fractional bundle from the left, of the allocation associated with the blue (resp. red, resp. green) vertex of Δ^* , i.e., x_1^* (resp. x_2^* , resp. x_3^*). Finally, as an illustrative example of left-first or right-first bundles, we observe that the central bundle is right-first in Δ^* .

Figure 3

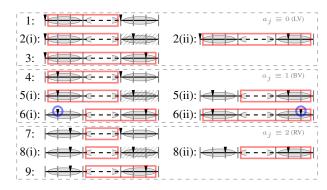


Figure 3: Given $j \in [n]$, we describe how the fractional bundle $\tilde{A}_j =$ $[a_j,b_j]$ of the main allocation of Δ^* can be rounded to obtain the integral bundle A_j in each of the following nine cases, assuming that A_{j+1}, \ldots, A_n have already been determined:

1: $a_j \equiv 0, b_j \equiv 0$: $A_j \leftarrow [a_j^-, b_j^-]$; 2: $a_j \equiv 0, b_j \equiv 1$: (i) $A_j \leftarrow \llbracket a_j^-, b_j^- \rrbracket$ if $b_i^+ \in A_{j+1}$, and (ii)

 $A_j \leftarrow \llbracket a_i^-, b_i^+ \rrbracket \text{ if } b_i^+ \not\in A_{j+1};$ 3: $a_j \equiv 0, b_j \equiv 2$: $A_j \leftarrow [a_j^-, b_j^+]$;

4: $a_j \equiv 1, b_j \equiv 0$: $A_j \leftarrow [a_j^-, b_j^-]$;

5: $a_j \equiv 1, b_j \equiv 1$: (i) $A_j \leftarrow \llbracket a_j^-, b_j^- \rrbracket$ if $b_j^+ \in A_{j+1}$, and (ii) $A_j \leftarrow [\![a_i^+, b_i^+]\!] \text{ if } b_i^+ \not\in A_{j+1};$

6: $a_j \equiv 1, b_j \equiv 2$: (i) $A_j \leftarrow [a_i^+, b_i^+]$ if A_j is left-first, and (ii) $A_j \leftarrow [a_j^-, b_j^+]$ if A_j is right-first;

 $\begin{array}{l} 7: \, a_j \equiv 2, b_j \equiv 0: A_j \leftarrow [\![a_j^+, b_j^-]\!]; \\ 8: \, a_j \ \equiv \ 2, b_j \ \equiv \ 1: \ (\mathrm{i}) \ A_j \ \leftarrow \ [\![a_j^+, b_j^-]\!] \ \ \mathrm{if} \ \ b_j^+ \ \in \ A_{j+1}, \ (\mathrm{ii}) \\ \end{array}$ $A_j \leftarrow \llbracket a_j^+, b_j^+ \rrbracket \text{ if } b_j^+ \not\in A_{j+1};$

9: $a_j \equiv 2, b_j \equiv 2$: $A_j \leftarrow [a_i^+, b_i^+]$.

The figure illustrates each of the nine cases as follows: each item and its three associated fractionality levels are represented by an ellipse divided into three parts (similarly to Figure 2 in Appendix G); the two black triangles in each case represent the positions $(a_j \text{ and } b_j)$ of the two knives that determine the j-th fractional bundle $\tilde{A}_j = [a_j, b_j]$ (by possibly cutting the two board items of the considered bundle); the red rectangle encloses all items that are fully included in the j-th bundle A_j of the rounded (integral) allocation A; the right-hand item, when marked with crossed lines, indicates that it was included in bundle A_{j+1} during the previous step of the rounding procedure; the blue circle on the left (resp. right) knife indicates that bundle $[a_j, b_j]$ is left-first (resp. right-first) in Λ^*

Figure 4

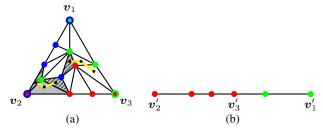


Figure 4: The figure on the left (a) illustrates an application of the Multi-coloring Sperner's lemma to a triangulation T of a 2-simplex $\Delta = \text{conv}(v_1, v_2, v_3)$. The multi-coloring function \mathcal{L} assigns a nonempty subset from {1 (Red), 2 (Green), 3 (Blue)} to each vertex $x \in V(T)$, and it is special; furthermore, in this specific case, \mathcal{L} satisfies the assumption done w.l.o.g. in the proof of Theorem 4.1, that is: $\mathcal{L}({m x}) = \{i \in [3]: {m x} \in F_i\}$ holds for any vertex ${m x}$ located on the boundary of Δ , and $|\mathcal{L}(x)| = 1$ holds for any internal vertex xnot located on the boundary. The four gray triangles represent the fullycolored simplices in T (w.r.t. the multi-coloring function \mathcal{L}); we observe that, differently from the standard Sperner's lemma, their number is not necessarily odd.

The minimal restriction $L:V(T)\to [3]$ of \mathcal{L} , as defined in the proof of Theorem 4.1, assigns the same color as ${\cal L}$ when that color is unique, and assigns an internal color when it is not (i.e., Green for v_1 , and Red for v_2 and v_3).

The figure on the right (b) illustrates the 1-simplex Δ' obtained from the instance of figure (a), following the steps outlined in the proof of Theorem 4.1. Δ' is homeomorphic to the topological space F' = $[v_2, v_3] \cup [v_2, v_3]$ obtained by the union of the 1-dimensional faces $[v_2, v_3]$ and $[v_3, v_1]$ of Δ . The figure also illustrates the triangulation (equivalent to) T' and the vertices of V(T') are colored following the minimal restriction L of \mathcal{L} .

Finally, the two yellow paths connecting some elementary 2-simplices in the left figure (a) form the graph G = (V, E) used in the proof of Theorem 4.1 to establish the existence of an odd number of fully-colored 2-simplices under the minimal restriction L, namely, the three gray triangles filled with black lines. These triangles are also fully colored with respect to the multi-coloring function \mathcal{L} ; we observe that, in this case, there are four such triangles, which are shown in gray.

Figure 5

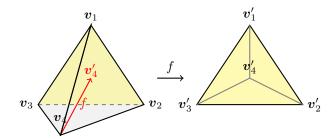


Figure 5: The figure illustrates the projection f, where Δ is a 3-simplex. The left image shows the original simplex Δ , along with the topological space of $F' = F_1 \cup F_2 \cup F_3$. Specifically, the 2-face $F_4 = \operatorname{conv}(\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3)$ opposite to \boldsymbol{v}_4 is represented with a yellow filling, while the other 2-faces F_1,F_2,F_3 (i.e., those constituting F') are filled with light gray. The right image depicts the 2-simplex Δ' , which is obtained by applying the projection f of F' onto F_4 . The red arrow on the left indicates the axis and orientation associated with the projection.

H **Pseudo-code of the Algorithms**

Algorithm 1: Dynamic Programming Algorithm (for Non-negative Valuations).

Input: An allocation instance $I = (N, M, (v_i)_{i \in N})$ with non-negative valuations.

Output: An EQ1P $_g^c$ allocation.

```
1: Initialize a set of integers C_v \leftarrow \{0\};
 2: for i = 1, ..., n do
        for l = 1, ..., m do
 3:
           for j = 1, \ldots, a do
 4:
               c \leftarrow v_i(\llbracket l, j \rrbracket) and add c to C_v;
 5:
           end for
 6:
 7:
        end for
 8: end for
 9: for c \in C_v do
        Initialize a \{1,\ldots,n\}\times\{0,\ldots,m\} boolean matrix
10:
        B, setting all entries to FALSE;
11:
        for j = 0, ..., m do
           if v_1^+([1,j]) \ge c \ge v_1^-([1,j]) then
12:
               B[1][j] \leftarrow \mathsf{TRUE};
13:
14:
           end if
        end for
15:
        for i=2,\ldots,n do
16:
17:
           for j = 0, ..., m do
               for \ell = 1, ..., j + 1 do
18:
19:
                  if v_i^+([\![\ell,j]\!]) \ge c \ge v_i^-([\![\ell,j]\!]) and
                     B[i-1][\ell-1] == \text{TRUE then}
                     B[i][j] \leftarrow \mathsf{TRUE};
20:
                     break;
21:
22:
                  end if
               end for
23:
24:
           end for
        end for
25:
        if B[n][m] ==TRUE then
26:
           break;
27:
28:
        end if
29: end for
30: Initialize \mathcal{A} = (A_1, \dots, A_n) arbitrarily;
31: j \leftarrow m;
32: for i = n, ..., 2 do
33:
        for \ell = 1, ..., j + 1 do
           if v_i^+(\llbracket\ell,j\rrbracket) \geq c \geq v_i^-(\llbracket\ell,j\rrbracket) and
34:
              B[i-1][\ell-1] == \text{TRUE then}
               A_i \leftarrow \llbracket \ell, j \rrbracket;
35:
               i \leftarrow \ell - 1
36:
              break;
37:
38:
           end if
        end for
39:
40: end for
41: A_1 \leftarrow [\![1,j]\!]
```

42: return \mathcal{A} ;

```
Algorithm 2: Local-search Alg. (for objective valuations).
```

Input: An allocation instance $I = (N, M = G \cup I)$ $C, (v_i)_{i \in N}$) with objective valuations.

Output: An EQX $_q^c$ allocation.

```
1: Initialize \mathcal{A} = (A_1, \dots, A_n) by setting A_1 \leftarrow M and
     A_i \leftarrow \emptyset for any i \in N \setminus \{1\};
 2: if v_1(A_1) > 0 then
        i \leftarrow n;
 3:
        while there exist j \in N and a good g \in A_j \cap G
        such that v_i(A_i) < v_j(A_j \setminus \{g\}) do
            A_i \leftarrow A_i \cup \{g\} \text{ and } A_i \leftarrow A_i \setminus \{g\};
 5:
            i \leftarrow \arg\min_{i \in N} v_i(A_i);
 6:
        end while
 7:
        return A;
 8:
 9: end if
10: if v_1(A_1) < 0 then
11:
        while there exist i \in N and a chore c \in A_i \cap C
12:
        such that v_i(A_i \setminus \{c\}) < v_j(A_j) do
            A_i \leftarrow A_i \setminus \{c\} \text{ and } A_j \leftarrow A_j \cup \{c\};
13:
14:
            j \leftarrow \arg\max_{i \in N} v_i(A_i);
        end while
15:
16: end if
17: return A;
```

Algorithm 3: Greedy Algorithm (for Objective Valuations).

Input: An allocation instance I = $(N, M = G \cup$ $C, (v_i)_{i \in N}$) with objective valuations.

Output: An EQ1 allocation.

```
1: Initialize the partial allocation \mathcal{A} = (A_1, \dots, A_n) by
     setting A_i \leftarrow \emptyset for any i \in N;
 2: for any good g \in G do
        let h \in \arg\min_{h \in N} v_h(A_h);
        A_h \leftarrow A_h \cup \{g\};
 4:
 5: end for
 6: for any chore c \in C do
        let h \in \arg \max_{h \in N} v_h(A_h);
 7:
 8:
        A_h \leftarrow A_h \cup \{c\};
 9: end for
10: return \mathcal{A};
```

Algorithm 4: Strongly-greedy Algorithm (for Objective Additive Valuations)

```
Input: An allocation instance I = (N, M = G \cup I)
C, (v_i)_{i \in N}) with additive objective valuations.
Output: An EQX_q^c allocation.
  1: Initialize the partial allocation \mathcal{A} = (A_1, \dots, A_n) by
     setting A_i \leftarrow \emptyset for any i \in N;
  2: initialize G' \leftarrow G and C' \leftarrow C;
  3: while G' \neq \emptyset do
        let h \in \arg\min_{h \in N} v_h(A_h);
        let g \in \arg \max_{g \in G'} v_h(\{g\});
        A_h \leftarrow A_h \cup \{g\}; G' \leftarrow G' \setminus \{g\};
  7: end while
  8: while C' \neq \emptyset do
         let h \in \arg \max_{h \in N} v_h(A_h);
  9:
         let c \in \operatorname{arg\,min}_{c \in C'} v_h(\{c\});
10:
        A_h \leftarrow A_h \cup \{c\}; C' \leftarrow C' \setminus \{c\};
```

12: end while13: return A;