Limiting Inequalities in Fair Division with Additive Value Preferences for Indivisible Social Items

Martin D. Aleksandrov[1],*

[1] *Freie Universität Berlin, Arnimallee 7, 14195 Berlin, Germany*

Abstract

We consider multi-agent item allocation problems where we differentiate between items that are good for all agents or bad for all agents. For this model with social items, we study six new properties that relax economic equity: three of them are based on the idea of removing one good or removing one bad; the other three of them are based on the idea of removing one good or duplicating one bad. We also give solutions for returning allocations of limited agent pairwise inequality, whenever agents have additive preferences for the items. Some of these solutions run in polynomial time, which makes them practical as opposed to standard intractable approaches such as minimizing the Gini index.

Keywords

Resource Allocation, Economic Equality, Algorithm Design

1. Introduction

Reducing inequalities and ensuring no one is left behind are integral to achieving the United Nations' Sustainable Development Goals. Inequality within and among countries is a persistent cause for concern. Despite some positive signs toward reducing inequality in some dimensions, such as reducing relative income inequality in some countries and preferential trade status benefiting lower-income countries, inequality still persists. Inequalities are also deepening for vulnerable populations in countries with weaker health systems and those facing existing humanitarian crises. Refugees and migrants, as well as indigenous peoples, older persons, people with disabilities, and children, are particularly at risk of being left behind. There are many obstacles when it comes to reducing inequalities. For example, we often do not perceive inequalities of others as a problem but rather as an opportunity for our own development. Also, we are often not fully aware of the main factors causing inequalities and the current mechanisms reducing inequalities, simply due to the fact that these factors and mechanisms are not as transparent and explainable as we would like them to be. As a consequence, we are often part of the problem and not part of the solution, thus compensating regularly for the successful efforts made by the United Nations in the direction of reducing inequalities. Indeed, we agree that it is hard to imagine a world without inequalities. But, can we imagine a world where inequalities are limited?

We do so in this paper in the context of a multi-agent resource allocation [1, 2, 3]. An extremely popular case for allocation is when the resource is represented by a collection of *indivisible* items. This is an appealing case when studying inequalities because most possessions in our life are indivisible. For example, if the neighbour has a car and we do not, we might be jealous of them. Furthermore, if they have a house and a car, our jealousy increases, but not as much as when we also have a house or a car. Perceiving lower inequalities might, therefore, relate to how comfortable we feel living in a given neighbourhood, district, and city and, consequently, to how well we perform in society. That is, reducing inequalities improves our well-being in the community!

Dividing private items is, however, not the only application in which we may want to reduce inequalities. Often, we may need to decide how to divide public (social) items among agents. For instance, consider two friends, say Anton and Bob, deciding what film to watch. Both of them obviously like films if they are deciding that. However, let us suppose that Anton prefers action films, whereas Bob prefers thriller films. When there is only a single decision to make, Nash [4] proposed maximizing the Nash welfare as an elegant solution to arrive at a mutually agreeable decision. However, in this single-decision setting, it might be impossible to make both Anton and Bob happy. By comparison, if we could get to make multiple decisions for multiple items, we might be able to reach a consensus by making sure that both Anton and Bob are happy with at least some of the decisions. For example, if they were to follow their movie with a beer, then Bob might be willing to agree to watch an action movie if he got to pick his favorite bar, and Anton might accept this compromise. Likewise, Anton might be willing to agree to watch a thriller movie if he got to pick his favorite bar, and Bob might accept this compromise.

AAAI 2023 Spring Symposia, Socially Responsible AI for Well-being, March 27–29, 2023, USA

*Corresponding author.

martin.aleksandrov@fu-berlin.de (M. D. Aleksandrov)
https://martofena.github.io/ (M. D. Aleksandrov)
0000-0003-0047-1235 (M. D. Aleksandrov)

Table 1
Symbol key: ✓ - the property can be achieved in all instances, × - the property may not be achieved in some instances, n - number of agents, m - number of items.

Property	**Goods and Bads**	**Goods**	**Bads**
DJFX$_0$ (additive)		× (by Theorem 1)	× (by Theorem 1)
JFX$_0$ (additive)		× (by Theorem 1)	× (by Theorem 1)
DJFX (additive)	✓, leximin++, $O(n^m)$ (Theorem 2)		
JFX (additive)	✓, Algorithm 1, $O(mn \log m)$ (Theorem 3)		
DJF1 (additive)	✓, Algorithm 2, $O(mn)$ (Theorem 4)		
JF1 (additive)	✓, Algorithm 1, $O(mn \log m)$ (by Theorem 3)		

Mathematically, this scenario can be modelled as having two indivisible items (a cinema location and a bar location) and two agents (Anton and Bob) who have additive utilities for the items (Anton has a fixed utility for the movie location but this utility increases if they get to pick the bar location as well). More generally, we consider a number of items and a number of agents. We let each item be either an *social good* (i.e. all agents weakly like it and perhaps some agents strictly like it) or an *social bad* (i.e. all agents weakly dislike it and perhaps some agents strictly dislike it) [5, 6]. We let each agent have *additive* utilities for bundles of items, which are sums of their utilities for the individual items in the bundles [7, 8].

Indivisible social items appear naturally in a number of real-world applications: (a) allocating research funds and their associated responsibilities among multiple research positions; (b) allocating paper tasks such as writing and editing, and their associated publication credits among researchers; (c) sharing credit, rent, and fare (http://www.spliddit.org/); (d) healthcare decisions regarding utility-attractiveness of therapies of varying effectiveness and societal benefit.

In such applications, after the agents submit their utilities, a (central) planner decides on a complete *allocation* that gives each item to an agent. Two common tasks of the planner are to compute allocations that minimise envy [9] or inequality [10] between the agents' utilities. While envy is a compelling notion in the private items setting, it makes less sense for social items. In our example, irrespective of where Anton and Bob go for watching a film, because they are watching the same film, it is not clear what it would mean for Anton to envy Bob and for Bob to envy Anton. If they could somehow trade places, they would still be at the same location, watching the same film, and not be any better off. Hence, their perceived fairness would be determined not by their subjective envy but by their objective satisfaction. This is also confirmed in another experimental study, where human subjects were asked to deliberate over an allocation of indivisible items and they picked outcomes minimised the inequalities far more often than they minimised the envy [11]. Thus, reducing inequalities can be a significant predictor of perceived fairness in practice.

Eliminating inequalities is a special case of reducing inequalities. An appealing axiomatic property that encodes the absence of inequalities is jealousy freeness [12]. An agent is *jealous* of another agent if the utility of the former agent for their own bundle is strictly lower than the utility of the latter agent for their own bundle. Otherwise, the former agent is *jealousy-free* of the latter agent. An allocation is *jealousy-free* if all agents derive equal utilities from their bundles.

In the setting with Anton and Bob, we can observe that jealousy-free allocations do not exist if there were just one indivisible item. As a response, we next investigate how we might relax jealousy freeness for limiting inequalities.

2. Limiting Inequalities

We propose three new relaxations of jealousy freeness. An allocation is *Jealousy-Free Up To Every Removed Item (JFX$_0$)* if it is jealousy-free and, otherwise, an agent who is jealous becomes jealousy-free of another agent, after any bad is removed from the jealous agent's bundle or any good is removed from the jealousy-free agent's bundle. Also, an allocation is *Jealousy-Free Up To Every Non-zero valued Removed Item (JFX)* whenever the removed item is non-zero valued, and *Jealousy-Free Up To Some Removed Item (JF1)* whenever the removal concerns some items. Further, we propose three alternatives to JFX$_0$, JFX, and JF1. An allocation is *Jealousy-Free Up To Every Removed or Duplicated Item (DJFX$_0$)* if each agent is jealousy-free of any other agent, otherwise, an agent who is jealous becomes jealousy-free of another agent, after any bad from the jealous agent bundle is duplicated to the jealousy-free agent's bundle or any good is removed from the jealousy-free agent's bundle. Also, an allocation is *Jealousy-Free Up To Every Non-zero valued Removed or Duplicated Item (DJFX)* whenever the removed or duplicated item is non-zero valued, and *Jealousy-Free Up To Some Removed or Duplicated Item (DJF1)* whenever the removal or duplication concerns some items. In this paper, we investigate the following question: *With additive utilities for indivisible social goods and bads, are there allocations of limited inequality in every instance?*

These properties relate to limiting inequalities, which is a task considered in economics for more than one hundred years [13]. However, unlike minimising the Gini index in an allocation, which is intractable [14], satisfying some of these properties is notably tractable. Table 1 contains our results. We show that DJFX$_0$ or JFX$_0$ allocations may not exist in some instances (see Theorem 1). Also, we give solutions (i.e. leximin++, Algorithms 1 and 2) for returning DJFX, JFX, DJF1, or JF1 allocations in every instance (see Theorems 2-4). As we can observe in Table 1, some of these solutions terminate in polynomial time. This enables limiting inequalities in various real-world resource allocation applications: see e.g. (a)-(d).

3. Related Works

In instances with strictly positive additive utilities, Gourvès et al. [12] considered computing *near jealousy-free* allocations. In such instances, an allocation is JFX$_0$ (JFX, DJFX$_0$, or DJFX) iff it is near jealousy-free. It follows that their approach returns such allocations in instances with goods. In contrast, we prove that DJFX$_0$ and JFX$_0$ allocations may not exist as soon as the utility functions are non-negative (see Theorem 1). With additive utilities, removing goods was used in [15] and removing bads was used in [16]. Freeman et al. used these techniques in isolation for defining *equitability up to every non-zero valued (some) item*. Unlike them, we combine these techniques for defining our properties. But, in instances with either goods or bads, an allocation is JFX (JF1) iff it is equitable up to every non-zero item (some item). In our instances with goods and bads, we prove that DJFX allocations and JFX allocations exist. In particular, we prove that the leximin++ solution [17] satisfies DJFX (see Theorem 2). Computing this solution relates to computing max-min fair allocations, which is shown to be intractable [18, 19]. By contrast, JFX allocations can be computed in a tractable manner (see Theorem 3). The same holds for DJF1 allocations (see Theorem 4).

4. Comparing Properties

We next compare the triple DJFX$_0$, DJFX, and DJF1, and the triple JFX$_0$, JFX, and JF1. For example, in general, each property limits from above the absolute inequality pairwise difference between agent utilities in an allocation. The limit is the maximum absolute agent utility for an item. So, we might be indifferent between the triples. But, in Example 1, we show that the former triple might be preferred to the latter triple. Indeed, achieving DJFX$_0$, DJFX, and DJF1 might optimise various welfares whereas achieving JFX$_0$, JFX, and JF1 might fail to do so. The planner could use this as a secondary criterion when deciding which triple of properties to use in practice.

Example 1. *Let us consider agents 1, 2, and 3 with utility profiles $(-\epsilon,-\epsilon,-\epsilon)$, $(-\epsilon,-\epsilon,-\epsilon)$, and $(-1,-1,-1)$, respectively. We let $\epsilon\rightarrow 0$ hold. Giving one item to each agent gives us a JFX$_0$, JFX, and JF1 allocation. We note that each such allocation: (1) minimises the product of agents' utilities (i.e. Nash welfare) to $-\epsilon^2$; (2) achieves a sum of agents' utilities (i.e. the utilitarian welfare) of $(-1-2\epsilon)$; (3) gives a difference between the minimum and maximum agent's utilities (i.e. the inequality difference) of $(-1+\epsilon)$. By comparison, an allocation, that shares the three items only among agents 1 and 2, maximises the Nash welfare to $0>-\epsilon^2$ and the utilitarian welfare to $-3\epsilon>(-1-2\epsilon)$, whilst minimising the inequality difference to $-2\epsilon>(-1+\epsilon)$.*

5. Relations to Responsible AI

Responsible AI is the practice of designing, developing, and deploying AI with good intentions to empower employees and businesses, and fairly impact customers and society. We respond to this by defining the six inequality properties DJFX$_0$, DJFX, DJF1, JFX$_0$, JFX, and JF1 that rely on the human emotion of jealousy and designing solutions for satisfying these properties, thus limiting this negative emotion. Like the Gini index, the properties provide measures of how humans might perceive fairness in practice. Unlike the Gini index, the properties operate on the individual level of each agent and not on the social level of all agents directly. Nevertheless, limiting the inequalities between the agent pairwise individual utility levels also limits naturally the social level and, therefore, the Gini index. From this perspective, the properties encode the well-being of individuals and society. This enables the implementation of individually and socially responsible AI for well-being in various application domains.

6. Application Domains

Limiting inequalities is a relevant task in various domains: in static domains like ours where the entire resource is fixed and available at one point in time; in dynamic domains where, for each out of multiple points in time, we have a static problem; in repeated domains such as allocating resources in multiple rounds. This work offers therefore a stepping stone for a better understanding of limiting inequalities in these domains. For example, in static domains, such allocations limit the agent pairwise difference in utility levels. Also, in dynamic domains, there are multiple points in time and, at each next point, we could bias the computation of allocations of the current resources with respect to (w.r.t.) the (aggregated) utility levels agents received in the previous points in time. Finally, in repeated domains, we can also do that w.r.t. the utility levels agents received in the previous rounds.

7. Formal Preliminaries

7.1. Model

We look at instances with $[n] = \{1, \ldots, n\}$ of $n \in \mathbb{N}_{\geq 2}$ agents and $[m] = \{1, \ldots, m\}$ of $m \in \mathbb{N}_{\geq 1}$ indivisible items. We let each $a \in [n]$ use a *utility function* $u_a : 2^{[m]} \to \mathbb{R}$. We let $u_a(\emptyset) = 0$ hold. We write $u_a(o)$ for $u_a(\{o\})$. For $M \subseteq [m]$, we say that $u_a(M)$ is *additive* iff $u_a(M) = \sum_{o \in M} u_a(o)$. The utilities are *identical* iff, for each $o \in [m]$, $u_a(o) = u_b(o)$ for each $a, b \in [n]$. We may write $u(o)$ in this case. Further, let us consider agent $a \in [n]$ and item $o \in [m]$. We say that o is *good* for a if $u_a(o) \geq 0$ holds. We refer to o as *pure good* whenever $u_a(o) > 0$ holds for each $a \in [n]$. Similarly, we say that o is *bad* for a if $u_a(o) \leq 0$ holds. We refer to o as *pure bad* whenever $u_a(o) < 0$ holds for each $a \in [n]$. We let $[m]$ be partitioned into *(social) goods* and *(social) bads*, respectively $G = \{o \in [m] | \forall a \in [n] : u_a(o) \geq 0\}$ and $B = \{o \in [m] | \forall a \in [n] : u_a(o) \leq 0\}$. A complete *allocation* $A = (A_1, \ldots, A_n)$ is such that (1) A_a is the bundle of agent $a \in [n]$, (2) $\bigcup_{a \in [n]} A_a = [m]$ holds, and (3) $A_a \cap A_b = \emptyset$ holds for each $a, b \in [n]$ with $a \neq b$.

7.2. Properties

Let us consider an allocation and a pair of agents, say a and b. One of them is jealousy-free of the other one, say b. Some of our notions rely on the idea of decreasing the utility of the agent who is jealousy-free, i.e. the b's utility.

Thus, agent a is DJFX$_0$ of agent b whenever a's utility is at least as much as b's utility, after *duplicating* each individual bad from a's bundle into b's bundle and *removing* each individual good from b's bundle.

Definition 1. (DJFX$_0$) *A is* DJFX$_0$ *if, $\forall a, b \in [n]$ s.t. agent a is not jealousy-free of agent b, (1) $\forall o \in A_a$ s.t. $u_a(A_a) \leq u_a(A_a) - u_a(o)$: $u_a(A_a) \geq u_b(A_b) + u_b(o)$ and (2) $\forall o \in A_b$ s.t. $u_b(A_b) \geq u_b(A_b) - u_b(o)$: $u_a(A_a) \geq u_b(A_b) - u_b(o)$.*

Also, agent a is DJFX of agent b whenever the above requirements hold for each individual non-zero bad in a's bundle, strictly increasing a's utility, and each individual non-zero good in b's bundle, strictly decreasing b's utility.

Definition 2. (DJFX) *A is* DJFX *if, $\forall a, b \in [n]$ s.t. agent a is not DJFX$_0$ of agent b, (1) $\forall o \in A_a$ s.t. $u_a(A_a) < u_a(A_a) - u_a(o)$: $u_a(A_a) \geq u_b(A_b) + u_b(o)$ and (2) $\forall o \in A_b$ s.t. $u_b(A_b) > u_b(A_b) - u_b(o)$: $u_a(A_a) \geq u_b(A_b) - u_b(o)$.*

Furthermore, agent a is DJF1 of agent b whenever the above requirements hold for some bad in a's bundle, strictly increasing a's utility, or some good in b's bundle, strictly decreasing b's utility.

Definition 3. (DJF1) *A is* DJF1 *if, $\forall a, b \in [n]$ s.t. agent a is not DJFX of agent b, (1) $\exists o \in A_a$ s.t. $u_a(A_a) \geq u_b(A_b) + u_b(o)$ or (2) $\exists o \in A_b$ s.t. $u_a(A_a) \geq u_b(A_b) - u_b(o)$.*

A DJFX$_0$ allocation satisfies further DJFX, i.e. DJFX$_0$ is *stronger* than DJFX. Also, DJFX is stronger than DJF1. These relations follow directly by the definitions of these axiomatic concepts.

In a similar fashion, we define JFX$_0$, JFX, and JF1 for a given allocation, where $u_a(A_a) \geq u_b(A_b) + u_b(o)$ in each of the definitions of DJFX$_0$, DJFX, and DJF1 is replaced with $u_a(A_a) - u_a(o) \geq u_b(A_b)$, respectively. JFX$_0$ is stronger than JFX and JFX is stronger than JF1.

Definition 4. (JFX$_0$) *A is* JFX$_0$ *if, $\forall a, b \in [n]$ s.t. agent a is not jealousy-free of agent b, (1)* $\forall o \in A_a$ s.t. $u_a(A_a) \leq u_a(A_a) - u_a(o)$: $u_a(A_a) - u_a(o) \geq u_b(A_b)$ and (2) $\forall o \in A_b$ s.t. $u_b(A_b) \geq u_b(A_b) - u_b(o)$: $u_a(A_a) \geq u_b(A_b) - u_b(o)$.*

Definition 5. (JFX) *A is* JFX *if, $\forall a, b \in [n]$ s.t. agent a is not JFX$_0$ of agent b, (1)* $\forall o \in A_a$ s.t. $u_a(A_a) < u_a(A_a) - u_a(o)$: $u_a(A_a) - u_a(o) \geq u_b(A_b)$ and (2) $\forall o \in A_b$ s.t. $u_b(A_b) > u_b(A_b) - u_b(o)$: $u_a(A_a) \geq u_b(A_b) - u_b(o)$.*

Definition 6. (JF1) *A is* JF1 *if, $\forall a, b \in [n]$ s.t. agent a is not JFX of agent b, (1)* $\exists o \in A_a$ s.t. $u_a(A_a) - u_a(o) \geq u_b(A_b)$ or (2) $\exists o \in A_b$ s.t. $u_a(A_a) \geq u_b(A_b) - u_b(o)$.*

By definition, DJFX$_0$, DJFX, and DJF1 coincide respectively with JFX$_0$, JFX, and JF1 in instances with just goods, whereas DJFX$_0$, DJFX, and DJF1 differ respectively from JFX$_0$, JFX, and JF1 in instances with bads: see Example 1.

8. Jealousy Freeness Up To Every Item

The strongest concepts DJFX$_0$ and JFX$_0$ might be violated by any allocation even in instances with identical additive utilities, which are often considered by [8]. The result holds whenever some of the items are valued with zero utilities.

Theorem 1. *There are instances with 2 agents and identical additive utilities, where* none *of the allocations satisfies DJFX$_0$ (JFX$_0$).*

Proof. We will show the result in an instance with goods and in an instance with bads, where there is one item that delivers zero utility to each agent. Such items might be good in cases when agents do not mind carrying another item (e.g. have space in the knapsack) with them but bad in cases when agents do mind carrying another item (e.g. do not have space in the knapsack) with them.

We first show the result with goods. Let us consider 2 agents, item g, and item b. We define the utilities as: $u_1(g) = u_2(g) = 1, u_1(b) = u_2(b) = 0$. By the symmetry of the utilities, we consider only two allocations: $A = (\{g\}, \{b\})$, $B = (\{g, b\}, \emptyset)$. We argue that none of these allocations is DJFX$_0$. To see this, we next give all violations of this property for each allocation: (1) $u_2(A_2) = 0 < 1 = u_1(A_1 \cup \{b\})$ and (2) $u_2(B_2) = 0 < 1 = u_1(B_1 \setminus \{b\})$. We argue in a similar way that none of the allocations is JFX$_0$. To see this, we observe the following violations of this property: (1) $u_2(A_2 \setminus \{b\}) = 0 < 1 = u_1(A_1)$ and (2) $u_2(B_2) = 0 < 1 = u_1(B_1 \setminus \{b\})$. The result follows.

We next show the result with bads. Let us consider 2 agents, item g, and item b. We define the utilities as: $u_1(g) = u_2(g) = -1, u_1(b) = u_2(b) = 0$. By the symmetry of the utilities, we consider only two allocations: $C = (\{g\}, \{b\})$, $D = (\{g, b\}, \emptyset)$. We argue that none of these allocations is DJFX$_0$. To see this, we next give all violations of this property for each allocation: (1) $u_1(C_1) = -1 < 0 = u_2(C_2 \setminus \{b\})$ and (2) $u_1(D_1) = -1 < 0 = u_2(D_2 \cup \{g\})$. We argue in a similar way that none of the allocations is JFX$_0$. To see this, we observe the following violations of this property: (1) $u_1(C_1) = -1 < 0 = u_2(C_2 \setminus \{b\})$ and (2) $u_1(D_1 \setminus \{b\}) = -1 < 0 = u_2(D_2)$. The result follows. This concludes the proof. □

In instances with additive pure goods, an allocation is DJFX$_0$ (JFX$_0$) iff it satisfies equitability up to every non-zero item [15]. Hence, such allocations exist in such instances. We conclude that adding bads to the instances ruins these properties.

9. Jealousy Freeness Up To Every Non-zero valued Item

By definition, DJFX$_0$ coincides with DJFX whenever the instance contains only pure goods and pure bads. This is not true in instances with goods and bads. In such instances, DJFX$_0$ allocations may not exist by Theorem 1 whereas DJFX allocations always exist by Theorem 2.

Such allocations are returned by the *leximin++* solution, proposed by [17]. To define it, we let $\overrightarrow{u}(A) \in \mathbb{R}^n$ denote the utility vector in A, which is (re-)arranged in some non-decreasing order. We write next $A \succ_{++} B$ if there exists an index $i \leq n$ such that $\overrightarrow{u}(A)_j = \overrightarrow{u}(B)_j$ and $|A_j| = |B_j|$ for each $1 \leq j < i$, and either $\overrightarrow{u}(A)_i > \overrightarrow{u}(B)_i$, or $\overrightarrow{u}(A)_i = \overrightarrow{u}(B)_i$ and $|A_i| > |B_i|$.

The leximin++ solution is defined as a maximal element under $\succ_{++}$. It maximises the least agent's utility, then maximises the bundle's size of an agent with the least utility, before it maximises the second least utility and the second least utility bundle's size, and so on.

Theorem 2. *In fair division with additive utilities for goods and bads, the leximin ++ solution could return in $O(m^n)$ time a DJFX allocation.*

Proof. Let A denote a leximin++ allocation. Suppose that A is not DJFX for a pair of agents $a, b \in [n]$ with $a \neq b$. That is, $u_a(A_a) < u_b(A_b)$. Also, (1) $u_a(A_a) < u_b(A_b) + u_b(o)$ holds for $o \in A_a$ with $u_a(A_a) < u_a(A_a) - u_a(o)$ or (2) $u_a(A_a) < u_b(A_b) - u_b(o)$ holds for $o \in A_b$ with $u_b(A_b) > u_b(A_b) - u_b(o)$. Wlog, let $u_1(A_1) \leq \ldots \leq u_n(A_n)$ denote the utility order induced by A. We let $k = \arg\max\{i \in [n] | u_i(A_i) \leq u_a(A_a)\}$. We note $a \leq k$ and $k < b$. We next consider two cases depending on whether condition (1) or condition (2) holds.

Case 1: Let (1) hold for bad $o \in A_a$. Let us move o from A_a to A_b. We let C denote this new allocation. That is, $C_a = A_a \setminus \{o\}$, $C_b = A_b \cup \{o\}$ and $C_c = A_c$ for each $c \in [n] \setminus \{a, b\}$. We argue that $C \succ_{++} A$ holds.

We note that $C_c = A_c$ holds for each $c \in [k] \setminus \{a\}$. We show $u_{a_k}(C_{a_k}) > u_k(A_k)$ where a_k is the kth agent in the utility order induced by C. If this agent is a, then $u_a(C_a) = u_a(A_a) - u_a(o) > u_a(A_a) = u_k(A_k)$ by (1). If this agent is b, then $u_b(C_b) = u_b(A_b) + u_b(o) > u_a(A_a) = u_k(A_k)$ by (1). Otherwise, $u_{a_k}(C_{a_k}) = u_{k+1}(A_{k+1}) > u_k(A_k)$ by the choice of k. It follows in each case that $u_d(C_d) \geq u_{a_k}(C_{a_k}) > u_k(A_k)$ holds for each agent $d \in [n] \setminus [([k] \setminus \{a\}) \cup \{a_k\}]$. Therefore, A cannot be leximin++.

Case 2: Let (2) hold for good $o \in A_b$. Let us move only item o from A_b to A_a. We let B denote this allocation: $B_a = A_a \cup \{o\}$, $B_b = A_b \setminus \{o\}$ and $B_c = A_c$ for each $c \in [n] \setminus \{a, b\}$. We next argue that $B \succ_{++} A$ holds.

As item o is good, it follows $u_a(B_a) \geq u_a(A_a)$. If $u_a(B_a) = u_a(A_a)$, then $B_c = A_c$ holds for each $c < a$, $|B_a| = |A_a| + 1$ and $B_d = A_d$ for each $d \in (a, k]$. Moreover, it follows that $u_e(B_e) > u_k(A_k) = u_a(A_a)$ holds for each agent $e \in [n] \setminus [k]$, including for agent b by (2). As $\succ_{++}$ maximises the bundle size as a secondary objective, it follows that $B \succ_{++} A$ holds. Hence, A cannot be the leximin++ solution. If $u_a(B_a) > u_a(A_a)$, then we can derive a contradiction as in Case 1 but we use in the proof condition (2) instead of condition (1). Hence, the leximin++ solution satisfies DJFX.

Finally, we conclude with the time complexity needed for computing the leximin++ solution. This solution can be computed by a simple naive brute-force algorithm that starts from a given allocation P and, for each allocation Q, checks whether $Q \succ_{++} P$ holds and if it holds continues with Q and, otherwise, with P. The allocation returned by this simple algorithm is maximal under the operator $\succ_{++}$ and, therefore, the leximin++ solution. The algorithm has to visit all $O(n^m)$ allocations (i.e. each out of m items can be allocated to each out of n agents) in the worst case. Therefore, it will terminate in $O(n^m)$ time at the latest. This concludes the proof. □

On the one hand, we found it challenging to guarantee DJFX in polynomial time for instances with additive utilities for goods and bads, even though intractable such allocations exist by Theorem 2. For this reason, we leave *open* this question. On the other hand, we can do this for JFX allocations in such instances. For this purpose, we give Algorithm 1.

Algorithm 1 first partitions the items into goods and bads. Until all goods are allocated, the algorithm then picks the least utility agent and gives them their most preferred remaining good. Until all bads are allocated, the algorithm then picks the greatest utility agent and gives them their least preferred remaining bad. This gives us a JFX allocation.

Algorithm 1 A JFX allocation with additive utilities.

```
procedure JFX-ADDITIVE($[n]$, $[m]$, $(u_a)_n$)
    $\forall a \in [n] : A_a \leftarrow \emptyset$
    $M^+ \leftarrow G$
    $M^- \leftarrow B$
    while $M^+ \neq \emptyset$ do                ▷ i.e. allocate all goods
        $b \leftarrow \arg\min_{a \in [n]} u_a(A_a)$
        $o \leftarrow \arg\max_{t \in M^+} u_b(t)$
        $A_b \leftarrow A_b \cup \{o\}$
        $M^+ \leftarrow M^+ \setminus \{o\}$
    while $M^- \neq \emptyset$ do                ▷ i.e. allocate all bads
        $c \leftarrow \arg\max_{a \in [n]} u_a(A_a)$
        $o \leftarrow \arg\min_{t \in M^-} u_c(t)$
        $A_c \leftarrow A_c \cup \{o\}$
        $M^- \leftarrow M^- \setminus \{o\}$
    return $A$
```

Theorem 3. *In fair division with additive utilities for goods and bads, Algorithm 1 returns in $O(mn \ln m)$ time a JFX allocation.*

Proof. Let us consider the partial allocation A. We assume that A is JFX and, under this assumption, we prove that extending A with the chosen item o preserves JFX. The result would then follow by the observation that the allocation when no item is allocated is JFX.

Case 1: If $o \in M^+$ is good for everyone, then $A_b = A_b \cup \{o\}$. Hence, $u_b(A_b) + u_b(o) \geq u_b(A_b)$. Moreover, we note that no agent has any bad in their bundle. This fact follows from the observation that the algorithm allocates all goods before any bads.

Sub-case 1.1 ($b \to a$): A is JFX. Therefore, $u_b(A_b) \geq u_a(A_a) - u_a(g^+)$ for each non-zero marginal good $g^+ \in A_a$. As $u_b(A_b) + u_b(o) \geq u_b(A_b)$, b remains JFX of a even after b receives o.

Sub-case 1.2 ($a \to b$): As b is a minimum utility agent in A, $u_a(A_a) \geq u_b(A_b)$. It immediately follows $u_a(A_a) \geq u_b(A_b) + u_b(o) - u_b(o)$. Let $g^+ \in A_b$ be a non-zero marginal good.

If $u_b(o) > u_b(g^+)$, then b would have picked o before g^+ by the fact that agents pick goods in a non-increasing utility order. Hence, it must be the case that each non-zero marginal good $g^+ \in A_b$ is such that $u_b(g^+) \geq u_b(o)$. This implies $u_b(A_b)+u_b(o)-u_b(o) \geq u_b(A_b)+u_b(o)-u_b(g^+)$ for each $g^+ \in A_b \cup \{o\}$. a remains JFX of b even after b receives o.

Case 2: If $o \in M^-$ is bad for everyone, then $A_c = A_c \cup \{o\}$. Hence, $u_c(A_c)+u_c(o) \leq u_c(A_c)$. Recall, $c \in [n]$ and $o \in M^-$ are such that $u_c(A_c)$ is maximum and $u_c(o)$ is minimum. We note that all goods are allocated at this point.

Sub-case 2.1 ($c \to a$): By the choice of c, $u_c(A_c) \geq u_a(A_a)$ holds for each $a \in [n]$. It follows $u_c(A_c) + u_c(o) - u_c(o) \geq u_a(A_a)$ and $u_c(A_c) \geq u_a(A_a) - u_a(g^+)$ for each non-zero marginal good $g^+ \in A_a$. Let $g^- \in A_c$ be a non-zero marginal bad.

If $u_c(o) < u_c(g^-)$, then c would have picked o before g^- by the fact that agents pick bads in a non-decreasing utility order. Hence, it must be the case that each non-zero marginal bad $g^- \in A_c$ is such that $u_c(g^-) \leq u_c(o)$. This implies $u_c(A_c) + u_c(o) - u_c(g^-) \geq u_c(A_c) + u_c(o) - u_c(o)$ for each $g^- \in A_c \cup \{o\}$. c remains JFX of a even after c receives o.

Sub-case 2.2 ($a \to c$): As A satisfies JFX, we have $u_a(A_a) + u_a(g^-) \geq u_c(A_c)$ for each non-zero bad $g^- \in A_a$ and $u_a(A_a) \geq u_c(A_c) + u_c(g^+)$ for each non-zero good $g^+ \in A_c$. But, as $u_c(o) \leq 0$, it follows $u_c(A_c) \geq u_c(A_c) + u_c(o)$ and $u_c(A_c) - u_c(g^+) \geq u_c(A_c) + u_c(o) - u_c(g^+)$. a remains JFX of c. This concludes the proof. □

The worst-case run time of Algorithm 1 is dominated by computing a ranking of the m items for each of the n agents. Computing such rankings can be done in $O(mn \ln m)$ time and space.

To sum up, we might prefer JFX to DJFX in instances with additive utilities because it remains unknown whether DJFX allocations are tractable in every instance, even though such allocations exist by Theorem 2.

10. Jealousy Freeness Up To Some Item

As we showed, the leximin++ solution is DJFX and it can be computed in $O(n^m)$ time. This might be fine for a constant m. However, m can be much larger than n in practice. For this reason, we may wish to return tractable DJF1 allocations. We can do this with Algorithm 2.

Algorithm 2 allocates the items one by one. If the current item is good, then the algorithm gives it to the least utility agent. Otherwise, the algorithm gives it to the greatest utility agent in a thought experiment, where a copy of the bad is allocated to every agent.

Algorithm 2 A DJF1 allocation with additive utilities.

```
procedure DJF1-ADDITIVE($[n], [m], (u_a)_n$)
    $\forall a \in [n] : A_a \leftarrow \emptyset$
    $M \leftarrow [m]$
    while $M \neq \emptyset$ do
        $o \leftarrow$ an item from $M$
        if $o$ is social good then
            $a \leftarrow \arg\min_{b \in [n]} u_b(A_b)$
        else                                ▷ i.e. $o$ is social bad
            $a \leftarrow \arg\max_{b \in [n]} u_b(A_b) + u_b(o)$
        $A_a \leftarrow A_a \cup \{o\}$
        $M \leftarrow M \setminus \{o\}$
    return $A$
```

Theorem 4. *In fair division with additive utilities for goods and bads, Algorithm 2 returns in $O(mn)$ time a DJF1 allocation.*

Proof. The proof is inductive. In the base case, no items are allocated. This is DJF1. In the hypothesis, we let A denote the partial allocation and assume that A is DJF1. In the step case, we show that allocating o to agent a preserves DJF1. By the hypothesis, agents $b \neq a$ and $c \neq a$ remain DJF1 after o is allocated to a because their allocations remain intact.

Case 1: Let o be a social good. In this case, $u_b(A_b) + u_b(o) \geq u_b(A_b)$ and $u_b(A_b) \geq u_a(A_a)$ for each $b \in [n]$. Let us consider some agent $c \neq a$. If $u_a(A_a) + u_a(o) > u_a(A_a)$, it follows that $u_c(A_c) \geq u_a(A_a) + u_a(o) - u_a(o)$ holds. If $u_a(A_a) + u_a(o) = u_a(A_a)$, it follows that $u_c(A_c) \geq u_a(A_a) + u_a(o)$ holds. c remains DJF1 of a.

In the opposite direction, as A is DJF1, we conclude $u_a(A_a) \geq u_c(A_c) - u_C(g^+)$ for some non-zero marginal good $g^+ \in A_c$ or $u_a(A_a) - u_a(g^-) \geq u_c(A_c)$ for some non-zero marginal bad $g^- \in A_a$. Thus, as item o is social good, we derive $u_a(A_a) + u_a(o) \geq u_a(A_a)$ and $u_a(A_a) + u_a(o) - u_a(g^-) \geq u_a(A_a) - u_a(g^-)$. a remains DJF1 of c.

Case 2: Let o be a social bad. In this case, $u_b(A_b) + u_b(o) \leq u_b(A_b)$ and $u_b(A_b) + u_b(o) \leq u_a(A_a) + u_a(o)$ for each $b \in [n]$. Let us consider some agent $c \neq a$. For the sake of contradiction, let us suppose that a is not DJF1 of c. This implies that $u_a(A_a) + u_a(o) < u_c(A_c) + u_c(o)$ must hold. But, this contradicts the choice of a. a remains DJF1 of agent c.

In the opposite direction, as A is DJF1, we conclude $u_c(A_c) \geq u_a(A_a) + u_a(g^+)$ for some non-zero marginal good $g^+ \in A_a$ or $u_c(A_c) \geq u_a(A_a) + u_a(g^-)$ for some non-zero marginal bad $g^- \in A_c$. Thus, as item o is social bad, we derive $u_a(A_a) + u_a(o) + u_a(g^-) \leq u_a(A_a) + u_a(g^-)$ and $u_a(A_a) + u_a(o) - u_a(g^+) \leq u_a(A_a) - u_a(g^+)$. Agent c remains DJF1 of agent a. □

The input of Algorithm 2 is bounded by $O(mn)$. Its running time is dominated by computing a sorting of the n agents' bundle utilities for each of the m items.

By Theorem 3, it follows that JFX allocations exist and, as by definition, such allocations satisfy JF1, it follows that Algorithm 1 returns JF1 allocations as well.

To sum up, we may be indifferent between DJF1 and JF1 in instances where agents have additive utilities for goods and bads, because both properties are tractable.

11. Future Directions

One future direction is to close the question we left open. For example, by Theorem 2, we know that intractable DJFX allocations exist in instances where agents have additive utilities for goods and bads, but we could not design a tractable algorithm that returns DJFX allocations in such instances. Another future direction is to study interactions of the proposed properties with economic efficiency criteria such as Pareto optimality. Finally, as social resources in practice are either public goods or public bads, we focused on instances with such items. However, in future work, we will also consider instances in which a fixed item can be good for some agents and bad for other agents.

12. Acknowledgements

Martin Aleksandrov was supported by the DFG Individual Research Grant on "Fairness and Efficiency in Emerging Vehicle Routing Problems" (497791398).

References

[1] S. J. Brams, A. D. Taylor, Fair Division - from Cake-cutting to Dispute Resolution, Cambridge University Press, 1996. URL: https://doi.org/10.1017/CBO9780511598975.

[2] H. Steinhaus, The problem of fair division, Econometrica 16 (1948) 101–104. URL: https://fu-berlin.primo.exlibrisgroup.com/discovery/openurl?institution=49KOBV_FUB&vid=49KOBV_FUB:FUB&rft.epage=104&rft.volume=16&url_ver=Z39.88-2004&rft.date=1948&rft.spage=101&rft.jtitle=Econometrica&rft.atitle=The%20problem%20of%20fair%20division&rft.title=Econometrica.

[3] H. P. Young, Equity: In Theory and Practice, Princeton University Press, 1995. URL: https://press.princeton.edu/books/paperback/9780691044644/equity.

[4] J. Nash, The bargaining problem, Econometrica 18 (1950) 155–162. URL: https://doi.org/10.1515/9781400829156-007.

[5] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, The efficiency of fair division, Theory of Computing Systems 50 (2012) 589–610. URL: https://doi.org/10.1007/s00224-011-9359-y.

[6] F. Sandomirskiy, E. Segal-Halevi, Efficient fair division with minimal sharing, 2020. URL: https://arxiv.org/abs/1908.01669. arXiv:1908.01669.

[7] H. Aziz, H. Moulin, F. Sandomirskiy, A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation, Operations Research Letters 48 (2020) 573–578. URL: https://www.sciencedirect.com/science/article/pii/S0167637720301024.

[8] H. Aziz, S. Rey, Almost group envy-free allocation of indivisible goods and chores, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI'20, 2021, pp. 39–45. URL: https://dl.acm.org/doi/abs/10.5555/3491440.3491446.

[9] D. K. Foley, Resource allocation and the public sector, Yale Economic Essays 7 (1967) 45–98. URL: https://www.proquest.com/docview/302230213?pq-origsite=gscholar&fromopenview=true.

[10] C. Gini, Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche. [Fasc. I.], Studi economico-giuridici pubblicati per cura della facoltà di Giurisprudenza della R. Università di Cagliari, Tipogr. di P. Cuppini, Bologna, 1912. URL: https://books.google.de/books?id=fqjaBPMxB9kC.

[11] D. Herreiner, C. Puppe, Envy freeness in experimental fair division problems, Theory and Decision 67 (2005). URL: https://doi.org/10.1007/s11238-007-9069-8.

[12] L. Gourvès, J. Monnot, L. Tlilane, Near fairness in matroids, in: Proceedings of the Twenty-First European Conference on Artificial Intelligence, ECAI'14, IOS Press, NLD, 2014, pp. 393–398. URL: https://dl.acm.org/doi/10.5555/3006652.3006719.

[13] N. Alon, Splitting necklaces, Advances in Mathematics 63 (1987) 247–253. URL: https://doi.org/10.1016/0001-8708(87)90055-7.

[14] M. Aleksandrov, C. Ge, T. Walsh, Fair division minimizing inequality, in: Progress in Artificial Intelligence, 19th EPIA Conference on AI, EPIA 2019, Vila Real, Portugal, September 3-6, 2019, Proceedings, Part II, 2019, pp. 593–605. URL: https://doi.org/10.1007/978-3-030-30244-3_49.

[15] R. Freeman, S. Sikdar, R. Vaish, L. Xia, Equitable allocations of indivisible goods, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI'19, AAAI Press, 2019, pp. 280–286. URL: https://dl.acm.org/doi/10.5555/3367032.3367073.

[16] R. Freeman, S. Sikdar, R. Vaish, L. Xia, Equitable allocations of indivisible chores, in: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '20, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2020, p. 384–392. URL: https://dl.acm.org/doi/10.5555/3398761.3398810.

[17] B. Plaut, T. Roughgarden, Almost envy-freeness with general valuations, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018, 2018, pp. 2584–2603. URL: https://doi.org/10.1137/19M124397X.

[18] I. Bezáková, V. Dani, Allocating indivisible goods, Association for Computing Machinery (ACM) SIGecom Exchanges 5 (2005) 11–18. URL: https://dl.acm.org/doi/10.1145/1120680.1120683.

[19] S. Dobzinski, J. Vondrák, Communication complexity of combinatorial auctions with submodular valuations, in: Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '13, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2013, pp. 1205–1215. URL: https://epubs.siam.org/doi/10.1137/1.9781611973105.87.