

The Frontier of Intractability for EFX with Two Agents

Paul W. Goldberg

University of Oxford

paul.goldberg@cs.ox.ac.uk

Kasper Høgh

Aarhus University

kh@cs.au.dk

Alexandros Hollender

EPFL

alexandros.hollender@epfl.ch

Abstract

We consider the problem of sharing a set of indivisible goods among agents in a fair manner, namely such that the allocation is envy-free up to any good (EFX). We focus on the problem of computing an EFX allocation in the two-agent case and characterize the computational complexity of the problem for most well-known valuation classes. We present a simple greedy algorithm that solves the problem when the agent valuations are weakly well-layered, a class which contains gross substitutes and budget-additive valuations. For the next largest valuation class we prove a negative result: the problem is PLS-complete for submodular valuations. All of our results also hold for the setting where there are many agents with identical valuations.

1 Introduction

The field of fair division studies the following fundamental question: given a set of resources, how should we divide them among a set of agents (who have subjective preferences over those resources) in a fair way? This question arises naturally in many settings, such as divorce settlement, division of inheritance, or dissolution of a business partnership, to name just a few. Although the motivation for studying this question is perhaps almost as old as humanity itself, the first mathematical investigation of the question dates back to the work of Banach, Knaster and Steinhaus [Steinhaus, 1948, 1949].

Of course, in order to study fair division problems, one has to define what exactly is meant by a *fair* division. Different fairness notions have been proposed to formalize this. Banach, Knaster and Steinhaus considered a notion which is known today as *proportionality*: every agent believes that it obtained at least a fraction $1/n$ of the total value available, where n is the number of agents. A generally¹ stronger notion, and one which seems more adapted to the motivating examples we mentioned above, is that of *envy-freeness* [Gamow and Stern, 1958; Foley, 1966; Varian, 1974]. A division of the resources is said to be envy-free, if no agent is envious, i.e., no agent values the bundle of resources obtained by some other agent strictly more than what it obtained itself.

As our motivating examples already suggest, the case with few agents – in fact, even just with two agents – is very relevant in practice. When the resources are divisible, such as for example money, water, oil, or time, the fair division problem with two agents admits a very simple and elegant solution: the cut-and-choose algorithm, which already appears in the Book of Genesis. As its name suggests, in the cut-and-choose algorithm one agent cuts the resources in half (according to its own valuation), and the other agent chooses its preferred piece, leaving the other piece to the first agent. It is easy to check that this guarantees envy-freeness, among other things. The case of divisible resources, which is usually called *cake cutting*, has been extensively studied for more than two agents. One of the main objectives in that line of research

¹As long as agents' valuations are subadditive, every envy-free division also satisfies proportionality.

can be summarized as follows: come up with approaches that achieve similar guarantees to cut-and-choose, but for more than two agents. This has been partially successful, and notable results include the proof of the existence of an envy-free allocation for any number of agents [Stromquist, 1980; Woodall, 1980; Su, 1999], as well as a finite, albeit very inefficient, protocol for computing one [Aziz and Mackenzie, 2016].

In many cases, however, assuming that the resources are divisible might be too strong an assumption. Indeed, some resources are inherently *indivisible*, such as a house, a car, or a company. Sometimes these resources can be made divisible by sharing them over time, for example, one agent can use the car over the week-end and the other agent on weekdays. But, in general, and in particular when agents are not on friendly terms with each other, as one would expect to often be the case for divorce settlements, this is not really an option.

Indivisible resources make the problem of finding a fair division more challenging. First of all, in contrast to the divisible setting, envy-free allocations are no longer guaranteed to exist. Indeed, this is easy to see even with just two agents and a single (indivisible) good that both agents would like to have. No matter who is given the good, the other agent will envy them. In order to address this issue of non-existence of a solution, various relaxations of envy-freeness have been proposed and studied in the literature. The strongest such relaxation, namely the one which seems closest to perfect envy-freeness, is called *envy-freeness up to any good* and is denoted by EFX [Caragiannis et al., 2019b; Gourvès et al., 2014]. An allocation is EFX if for all agents i and j , agent i does not envy agent j , after removal of *any* single good from agent j 's bundle. In other words, an allocation is *not* EFX, if and only if there exist agents i and j , and a good in j 's bundle, so that i envies j 's bundle even after removal of that good.

For this relaxed notion of envy-freeness, it is possible to recover existence, at least in some cases. An EFX allocation is guaranteed to exist for two agents with any monotone valuations [Plaut and Roughgarden, 2020], and for three agents if we restrict the valuations to be additive [Chaudhury et al., 2020]. It is currently unknown whether it always exists for four or more agents, even just for additive valuations.

Surprisingly, proving the existence of EFX allocations for two agents is non-trivial. In order to use the cut-and-choose approach, we need to be able to “cut in half”. In the divisible setting, this is straightforward. But, in the indivisible setting, we need to “cut in half in the EFX sense,” i.e., divide the goods into two bundles such that the first agent is EFX with either bundle. In other words, we first need to show the existence of EFX allocations for two identical agents, namely two agents who share the same valuation function, which is not a trivial task.

Plaut and Roughgarden [2020] provided a solution to this problem by introducing the *leximin++ solution*. Given a monotone valuation function, they defined a total ordering over all allocations called the leximin++ ordering. They proved that for two identical agents, the leximin++ solution, namely the global maximum with respect to the leximin++ ordering, must be an EFX allocation. As mentioned above, using the cut-and-choose algorithm, this shows the existence of EFX allocations for two, possibly different, agents. Unfortunately, computing the leximin++ solution is computationally intractable² and so, while this argument proves the existence of EFX allocations, it does not yield an efficient algorithm.

Nevertheless, for two agents with *additive* valuations, Plaut and Roughgarden [2020] provided a polynomial-time algorithm based on a modification of the Envy-Cycle elimination algorithm of Lipton et al. [2004]. They also provided a lower bound for the problem in the more general class of submodular valuations, but not in terms of computational complexity (i.e., not in the standard Turing machine model). Namely, they proved that for two identical agents with submodular valuations computing an EFX allocation requires an exponential number of queries in the query complexity model.

²Computing the leximin++ solution is NP-hard, even for two identical agents with additive valuations. This can be shown by a reduction from the PARTITION problem (see [Plaut and Roughgarden, 2020, Footnote 7] and note that the argument also applies to leximin++).

Their work naturally raises the following two questions about the problem of computing an EFX allocation for two agents:

1. What is the *computational* complexity of the problem for submodular valuations?
2. What is the computational complexity of the problem for well-known valuation classes lying between additive and submodular,³ such as gross substitutes, OXS, and budget-additive?

Note that it does not make sense to study the query complexity for additive valuations, since a polynomial number of queries is sufficient to reconstruct the whole valuation functions (and the amount of computation then needed to determine a solution is not measured in the query complexity). However, it does make sense to study the computational complexity of the problem for submodular valuations, as well as other classes beyond additive. The query lower bound by Plaut and Roughgarden essentially says that many queries are needed in order to gather enough information about the submodular valuation function to be able to construct an EFX allocation. But it does not say anything about the *computational* hardness of finding an EFX allocation. Their lower bound does not exclude the possibility of a polynomial-time algorithm for submodular valuations in the standard Turing machine model. Studying the problem in the computational complexity model allows us to investigate how hard it is to solve when the valuation functions are given in some succinct representation, e.g., as a few lines of code, or any other form that allows for efficient evaluation.

Our contribution. We answer both of the aforementioned questions:

1. For submodular valuations, we prove that the problem is PLS-complete in the standard Turing machine model, even with two identical agents.
2. We present a simple greedy algorithm that finds an EFX allocation in polynomial time for two agents with *weakly well-layered* valuations, a class of valuation functions that we define in this paper and which contains all well-known strict subclasses of submodular, such as gross substitutes, OXS, and budget-additive.

Together, these two results completely resolve the computational complexity of the problem for all valuation classes in the standard complement-free hierarchy ($\text{additive} \subseteq \text{OXS} \subseteq \text{gross substitutes} \subseteq \text{submodular} \subseteq \text{XOS} \subseteq \text{subadditive}$) introduced by Lehmann et al. [2006]. Furthermore, just like in the work of Plaut and Roughgarden [2020], our negative and positive results also hold for any number of *identical* agents.

Regarding the PLS-completeness result, the membership in PLS is easy to show using the leximin++ ordering of Plaut and Roughgarden [2020]. The PLS-hardness is more challenging. The first step of our hardness reduction is essentially identical to the first step in the corresponding query lower bound of Plaut and Roughgarden [2020]: a reduction from a local optimization problem on the Kneser graph to the problem of finding an EFX allocation. The second step of the reduction is our main technical contribution: we prove that finding a local optimum on a Kneser graph is PLS-hard⁴, which might be of independent interest.

³In particular, Plaut and Roughgarden [2020, Section 7] propose studying the complexity of fair division problems with respect to the hierarchy of complement-free valuations ($\text{additive} \subseteq \text{OXS} \subseteq \text{gross substitutes} \subseteq \text{submodular} \subseteq \text{XOS} \subseteq \text{subadditive}$) introduced by Lehmann et al. [2006].

⁴We note that proving a tight computational complexity lower bound is more challenging than proving a query lower bound, because we have to reduce from problems with more structure. Indeed, the exponential query lower bound for the Kneser problem (and thus also for the EFX problem) can easily be obtained as a byproduct of our reduction.

Further related work. The existence and computation of EFX allocations has been studied in various different settings, such as for restricted versions of valuation classes [Amanatidis et al., 2021; Babaioff et al., 2021], when some items can be discarded [Caragiannis et al., 2019a; Chaudhury et al., 2021b; Berger et al., 2022; Chaudhury et al., 2021a], or when valuations are drawn randomly from a distribution [Manurangsi and Suksompong, 2021].

A weaker relaxation of envy-freeness is *envy-freeness up to one good* (EF1) [Budish, 2011; Lipton et al., 2004]. It can be computed efficiently for any number of agents with monotone valuations using the Envy-Cycle elimination algorithm [Lipton et al., 2004]. If one is also interested in economic efficiency, then it is possible to obtain an allocation that is both EF1 and Pareto-optimal in pseudopolynomial time for additive valuations [Barman et al., 2018]. For more details about fair division of indivisible items, we refer to the two recent surveys by Amanatidis et al. [2022] and Aziz et al. [2022].

Outline. We begin with Section 2 where we formally define the problem and solution concept, as well as some standard valuation classes of interest. In Section 3 we introduce *weakly well-layered* valuation functions, and present our simple greedy algorithm for computing EFX allocations. Finally, in Section 4 we prove our main technical result, the PLS-completeness for submodular valuations.

2 Preliminaries

We consider the problem of discrete fair division where an instance consists of a set of agents N , a set of goods M , and for every agent $i \in N$ a valuation function $v_i: 2^M \rightarrow \mathbb{R}_{\geq 0}$ assigning values to bundles of goods. All valuation functions will be assumed to be *monotone*, meaning that for any subsets $S \subseteq T \subseteq M$ it holds that $v(S) \leq v(T)$, and *normalized*, i.e., $v(\emptyset) = 0$.

We now introduce the different types of valuation functions that are of interest to us. A valuation $v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ is *additive* if $v(S) = \sum_{g \in S} v(\{g\})$ for every $S \subseteq M$. The hardness result we present in Section 4 holds for *submodular* valuations. These are valuations that satisfy the following diminishing returns condition that whenever $S \subseteq T$ and $x \notin T$ it holds that $v(S \cup \{x\}) - v(S) \geq v(T \cup \{x\}) - v(T)$.

Next, for our results in the positive direction, we introduce the classes of *gross substitutes* and *budget-additive* valuations, both contained in the class of submodular valuations. Before defining gross substitutes valuations, we have to introduce some notation. For a price vector $p \in \mathbb{R}^m$ on the set of goods, where $m = |M|$, the function v_p is defined by $v_p(S) = v(S) - \sum_{g \in S} p_g$ for any subset $S \subseteq M$, and the demand set is $D(v, p) = \arg \max_{S \subseteq M} v_p(S)$. A valuation v is *gross substitutes* if for any price vectors $p, p' \in \mathbb{R}^m$ with $p \leq p'$ (meaning that $p_g \leq p'_g$ for all $g \in M$), it holds that if $S \in D(v, p)$, then there exists a demanded set $S' \in D(v, p')$ such that $\{g \in S: p_g = p'_g\} \subseteq S'$. That is to say, if some good g is demanded at prices p and the prices of some *other* goods increase, then g will still be demanded. These valuations have various nice properties, for instance guaranteeing existence of Walrasian equilibria [Gul and Stacchetti, 1999]. Lastly, a valuation v is *budget-additive* if it is of the form $v(S) = \min\{B, \sum_{g \in S} w_g\}$ for reals $B, w_1, \dots, w_m \geq 0$. [Lehmann et al., 2006] show that a budget-additive valuation need not satisfy the gross substitutes condition. See Figure 1 for the relationship between the valuation classes.

Envy-freeness up to any good (EFX). The goal of fair division is to find an allocation of the goods to the agents (i.e., a partitioning $M = X_1 \sqcup \dots \sqcup X_n$) satisfying some notion of fairness. One might hope for an *envy-free* division in which every agent prefers his own bundle over the bundle of any other agent, that is, $v_i(X_i) \geq v_i(X_j)$ for all $i, j \in N$. Such a division need not exist, however, as can be seen in the case where one has to divide one good among two agents, as already mentioned in the introduction. Therefore various weaker notions of fairness have been

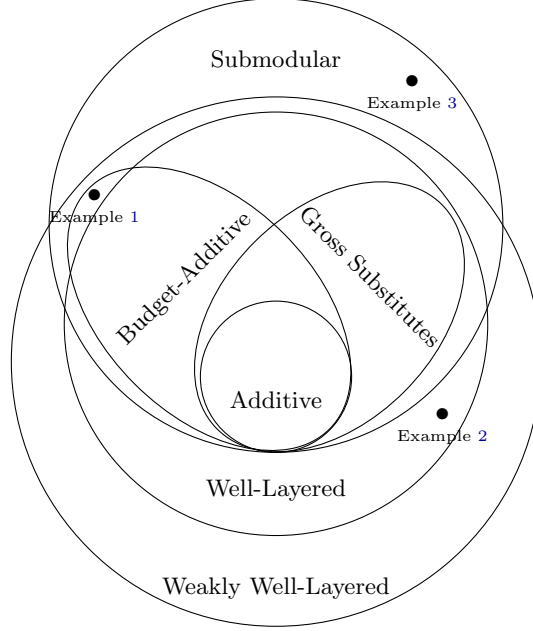


Figure 1: Inclusions of valuation classes

studied. In this paper we consider the notion of *envy-freeness up to any good* (EFX) introduced by Caragiannis et al. [2019b], and before that by Gourvès et al. [2014] under a different name. An allocation (X_1, \dots, X_n) is said to be EFX if for any $i, j \in N$ and any $g \in X_j$ it holds that $v_i(X_i) \geq v_i(X_j \setminus \{g\})$.

3 Polynomial-time Algorithm for Weakly Well-Layered Valuations

In this section we present our positive result, namely the polynomial-time algorithm for computing an EFX allocation for two agents with weakly well-layered valuations. To be more precise, our algorithm works for any number of agents that all share the same weakly well-layered valuation function. As a result, using cut-and-choose it can then be used to solve the problem with two possibly *different* agents. We begin with the definition of this new class of valuations, and then present the algorithm and prove its correctness.

3.1 Weakly Well-Layered Valuations

We introduce a property of valuation functions and situate this with respect to well-known classes of valuation functions.

Definition 3.1. A valuation function $v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ is said to be *weakly well-layered* if for any $M' \subseteq M$ the sets S_0, S_1, S_2, \dots obtained by the greedy algorithm (that is, $S_0 = \emptyset$ and $S_i = S_{i-1} \cup \{x_i\}$ where $x_i \in \arg \max_{x \in M' \setminus S_{i-1}} v(S_{i-1} \cup \{x\})$ for $1 \leq i \leq |M'|$) are optimal in the sense that $v(S_i) = \max_{S \subseteq M': |S|=i} v(S)$ for all i .

We can reformulate this definition as follows: a valuation function v is weakly well-layered if and only if, for all $M' \subseteq M$ and all i , the optimization problem

$$\begin{aligned} \max \quad & v(S) \\ \text{s.t.} \quad & S \subseteq M' \\ & |S| \leq i \end{aligned} \tag{1}$$

can be solved by using the natural greedy algorithm. Note that since we only consider monotone valuations, we can also use the condition $|S| = i$ instead of $|S| \leq i$.

The reformulation of the definition in terms of the optimization problem (1) is reminiscent of one of the alternative definitions of a matroid. Consider the optimization problem

$$\begin{aligned} \max \quad & v(S) \\ \text{s.t.} \quad & S \in \mathcal{F} \end{aligned} \tag{2}$$

where $v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ is a valuation function and \mathcal{F} is an independence system on M . Then, it is well-known that \mathcal{F} is a matroid, if and only if, for all additive valuations v , the optimization problem (2) can be solved by the natural greedy algorithm [Rado, 1957; Gale, 1968; Edmonds, 1971]. In other words, the class of set systems (namely, matroids) is defined by fixing a class of valuations (namely, additive). The alternative definition of weakly well-layered valuations given in (1) can be viewed as doing the opposite: the class of valuations (namely, weakly well-layered) is defined by fixing a class of set systems (namely, all uniform matroids on subsets $M' \subseteq M$, or, more formally, $\mathcal{F} = \{S \subseteq M' : |S| \leq i\}$ for all $M' \subseteq M$ and all i).

Relationship to other valuation classes. We begin by showing that any gross substitutes valuation is weakly well-layered. In particular, this also implies that OXS valuations, which are a special case of gross substitutes, are also weakly well-layered. Paes Leme [2017] proved that gross substitutes valuation functions satisfy the stronger condition of being *well-layered*, that is, for any $p \in \mathbb{R}^m$ it holds that if S_0, S_1, S_2, \dots is constructed greedily with respect to the valuation v_p , where $v_p(S) := v(S) - \sum_{g \in S} p_g$, then S_i satisfies that $S_i \in \arg \max_{S \subseteq M: |S|=i} v_p(S)$.

Lemma 1. *If $v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ is well-layered, then it is also weakly well-layered. In particular, gross substitutes valuations are weakly well-layered.*

Proof. Assume that $v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ is well-layered and let $M' \subseteq M$. Assume that the sequence S_0, S_1, S_2, \dots is constructed via the greedy algorithm: that is $S_0 = \emptyset$ and $S_i = S_{i-1} \cup \{x_i\}$ where $x_i \in \arg \max_{x \in M' \setminus S_{i-1}} v(S_{i-1} \cup \{x\})$ for $1 \leq i \leq |M'|$. We have to show that $v(S_i) = \max_{S \subseteq M': |S|=i} v(S)$.

In order to exploit the assumption that v is well-layered, we introduce a price vector $p \in \mathbb{R}^m$ given by

$$p_g = \begin{cases} 0 & g \in M' \\ v(M) + 1 & g \notin M' \end{cases}$$

One sees that the sequence S_0, S_1, S_2, \dots can occur via the greedy algorithm for the valuation v_p , because goods not in M' cannot be chosen as their prices are too high. As v is well-layered, we have that $v_p(S_i) = \max_{S \subseteq M: |S|=i} v_p(S)$. As $p_g = 0$ for all $g \in M'$, this implies that $v(S_i) = \max_{S \subseteq M': |S|=i} v(S)$. We conclude that v is weakly well-layered. \square

Closure property. We note that the class of weakly well-layered valuations is closed under composition with a strictly increasing function.

Lemma 2. *Let $v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ be weakly well-layered and let $f: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ strictly increasing. Then the composition $f \circ v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ is weakly well-layered.*

Proof. Let $M' \subseteq M$ and assume that S_0, S_1, S_2, \dots are constructed greedily, that is $S_0 = \emptyset$ and $S_i = S_{i-1} \cup \{x_i\}$ where $x_i \in \arg \max_{x \in M' \setminus S_{i-1}} f(v(S_{i-1} \cup \{x\}))$ for $1 \leq i \leq |M'|$. As f is strictly increasing, we see that $x_i \in \arg \max_{x \in M'} f(v(S_{i-1} \cup \{x\}))$ if and only if $x_i \in \arg \max_{x \in M'} v(S_{i-1} \cup \{x\})$. Therefore S_0, S_1, S_2, \dots could also arise via the greedy construction based on the valuation

v . As v is weakly well-layered, this implies that $v(S_i) = \max_{S \subseteq M': |S|=i} v(S)$ for all i . As f is increasing, this shows that $f(v(S_i)) = \max_{S \subseteq M': |S|=i} f(v(S))$ for all i . We conclude that $f \circ v$ is weakly well-layered. \square

Lemma 3. *Let $v: 2^M \rightarrow \mathbb{R}_{\geq 0}$ be weakly well-layered and $B \geq 0$. Then the valuation $u: 2^M \rightarrow \mathbb{R}_{\geq 0}$ given by $u(S) = \min(v(S), B)$ is weakly well-layered.*

Proof. Let S_0, S_1, S_2, \dots be constructed greedily from the valuation u . Suppose that S_0, S_1, \dots, S_k have utility $< B$ and that S_{k+1}, S_{k+2}, \dots have utility B . As $x \mapsto \min(x, B)$ is strictly increasing on $[0, B)$, the sets S_0, S_1, \dots, S_k could have been constructed greedily from v . As v is weakly well-layered, they are therefore optimal of their given size for v and therefore also for u . The sets S_{k+1}, \dots all have maximal utility B and are therefore optimal of their given sizes. \square

As a corollary it follows that the class of budget-additive valuations satisfies the weakly well-layered property.

Corollary 1. *Any budget-additive valuation is weakly well-layered.*

In contrast, we note that a budget-additive valuation is not necessarily well-layered, as the following example shows.

Example 1. Consider the budget-additive valuation on three goods a, b, c with values $v_a = v_b = 2$, $v_c = 4$ and a budget of $B = 4$. Let $p = (1, 1, 2)$ be a price vector. Under these prices, the greedy algorithm would pick good c as its first item. However, $\{a, b\}$ is the unique optimal bundle of size 2, and so the greedy algorithm would fail in this case. As a result, the valuation is not well-layered.

The results of this subsection are summarised in Figure 1. Note also that the classes of submodular valuations and weakly well-layered valuations are incomparable. For an example of a valuation function that is submodular but not weakly well-layered, see Example 3 in the next section. For the other direction, see the following example of a valuation that is well-layered (and thus weakly well-layered), but not submodular.

Example 2. Consider the valuation function v on two goods a, b given by $v(\{a, b\}) = 1$ and $v(\emptyset) = v(\{a\}) = v(\{b\}) = 0$. This valuation function is seen to be well-layered (and thus weakly well-layered), because subsets of equal size have the same valuation. However, it is not submodular, because $v(\{a\} \cup \{b\}) - v(\{a\}) = 1 > 0 = v(\emptyset \cup \{b\}) - v(\emptyset)$.

3.2 The Greedy EFX Algorithm

We now present a simple algorithm that computes an EFX allocation for many agents that all share the same weakly well-layered valuation function v .

Theorem 3.1. *If the valuation function v is weakly well-layered, then the output of Algorithm 1 is EFX. In particular, by using the cut-and-choose protocol one may compute an EFX allocation for two agents with different valuations as long as one of these valuations is weakly well-layered.*

Proof. We show that the algorithm maintains a partial EFX allocation throughout. Initially the partial allocation is empty and so clearly EFX. Suppose that at the beginning of some round the current partial allocation (X_1, \dots, X_n) is EFX and that some agent $i \in N$ receives a good g in this round. We have to show that the new (partial) allocation (X'_1, \dots, X'_n) is EFX, where $X'_i = X_i \cup \{g\}$ and $X'_j = X_j$ for $j \neq i$. Clearly, the only thing we have to argue is that $v(X'_i \setminus \{g'\}) \leq v(X'_j)$ for all $j \in N$ and all $g' \in X'_i$. As i received a good in the current round we have that $v(X_i) \leq v(X_j) = v(X'_j)$. Therefore, it suffices to argue that $v(X'_i \setminus \{g'\}) \leq v(X_i)$ for all $g' \in X'_i$. This last inequality follows from v being weakly well-layered by taking $M' = X'_i$.

Algorithm 1 Greedy EFX

Input: N, M, v **Output:** EFX allocationLet $A_i = \emptyset$ for $i \in N$.Let $R = M$.**while** $R \neq \emptyset$ **do** $i = \arg \min_{j \in N} v(A_j)$ $g = \arg \max_{x \in R} v(A_i \cup \{x\})$ $A_i = A_i \cup \{g\}$ $R = R \setminus \{g\}$ **end while****return** (A_1, \dots, A_n)

With this M' , the set X_i could namely be produced by running the greedy algorithm. Therefore, X_i is an optimal subset of $M' = X'_i$ of size $|X_i| = |X'_i| - 1$, meaning that $v(X'_i \setminus \{g'\}) \leq v(X_i)$ for all $g \in X'_i$. \square

The algorithm can fail to provide an EFX allocation for submodular valuations that are not weakly well-layered, as the following example shows.

Example 3. Consider an instance with two agents and four goods denoted a, b, c, d , where the valuation function v is given by: $v(\{a\}) = 11, v(\{b\}) = v(\{c\}) = 10, v(\{d\}) = 16, v(\{a, b\}) = 15, v(\{a, c\}) = 15, v(\{b, c\}) = 17, v(\{a, b, c\}) = 18$, and $v(S) = 18$ for all sets S that satisfy $d \in S$ and $|S| \geq 2$. It can be checked by direct computation that v is indeed submodular. The greedy EFX algorithm yields: agent 1 gets good d , and then agent 2 gets goods a, b, c . This allocation is not EFX, because $v(\{d\}) < v(\{b, c\})$.

4 PLS-completeness for Submodular Valuations

Total NP search problems (TFNP). A total search problem is given by a relation $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$ that satisfies: for all $x \in \{0, 1\}^*$, there exists $y \in \{0, 1\}^*$ such that $(x, y) \in R$. The relation R is interpreted as the following computational problem: given $x \in \{0, 1\}^*$, find some $y \in \{0, 1\}^*$ such that $(x, y) \in R$. The class TFNP [Megiddo and Papadimitriou, 1991] is defined as the set of all total search problems R such that the relation R is polynomial-time decidable (i.e., given some x, y we can check in polynomial time whether $(x, y) \in R$) and polynomially balanced (i.e., there exists some polynomial p such that $|y| \leq p(|x|)$ whenever $(x, y) \in R$).

Let R and S be two problems in TFNP. We say that R reduces to S if there exist polynomial-time functions $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ and $g : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ such that for all $x, y \in \{0, 1\}^*$: if $(f(x), y) \in S$, then $(x, g(y, x)) \in R$. In other words, f maps an instance of R to an instance of S , and g maps back any solution of the S -instance to a solution of the R -instance.

Polynomial Local Search (PLS). Johnson et al. [1988] introduced the class PLS, a subclass of TFNP, to capture the complexity of computing locally optimal solutions in settings where local improvements can be computed in polynomial time. In order to define the class PLS, we proceed as follows: first, we define a set of basic PLS problems, and then define the class PLS as the set of all TFNP problems that reduce to a basic PLS problem.

A *local search problem* Π is defined as follows. For every instance⁵ $I \in \{0, 1\}^*$, there is a finite set $F_I \subseteq \{0, 1\}^*$ of *feasible solutions*, an objective function $c_I : F_I \rightarrow \mathbb{N}$, and for every

⁵A more general definition would also include a polynomial-time recognizable set $D_\Pi \subseteq \{0, 1\}^*$ of valid instances. The assumption that $D_\Pi = \{0, 1\}^*$ is essentially without loss of generality. Indeed, for $I \notin D_\Pi$ we can define $F_I = \{0\}$, $c_I(0) = 1$ and $N_I(0) = \{0\}$. Note that this does not change the complexity of the problem.

feasible solution $s \in F_I$ there is a neighborhood $N_I(s) \subseteq F_I$. Given an instance I , one seeks a *local optimum* $s^* \in F_I$ with respect to c_I and N_I , meaning, in case of a maximization problem, that $c_I(s^*) \geq c_I(s)$ for all neighbors $s \in N_I(s^*)$.

Definition 4.1. A local search problem Π is a basic PLS problem if there exists some polynomial p such that $F_I \subseteq \{0, 1\}^{p(|I|)}$ for all instances I , and if there exist polynomial-time algorithms A, B and C such that:

1. Given an instance I , algorithm A produces an initial feasible solution $s_0 \in F_I$.
2. Given an instance I and a string $s \in \{0, 1\}^{p(|I|)}$, algorithm B determines whether s is a feasible solution and, if so, computes the objective value $c_I(s)$.
3. Given an instance I and any feasible solution $s \in F_I$, the algorithm C checks if s is locally optimal and, if not, produces a feasible solution $s' \in N_I(s)$ that improves the objective value.

Note that any basic PLS problem lies in TFNP.

Definition 4.2. The class PLS is defined as the set of all TFNP problems that reduce to a basic PLS problem.

A problem is PLS-complete if it lies in PLS and if every problem in PLS reduces to it. [Johnson et al. \[1988\]](#) showed that the so-called FLIP problem is PLS-complete. We will define this problem later when we make use of it to prove our PLS-hardness result.

4.1 PLS-membership

[Plaut and Roughgarden \[2020\]](#) prove the existence of an EFX allocation when all agents share the same monotone valuation, by introducing the leximin++ solution. In this section, we show how their existence proof can be translated into a proof of PLS-membership for the following problem.

Definition 4.3 (IDENTICAL-EFX). An instance $I = (N, M, C)$ of the IDENTICAL-EFX search problem consists of a set of agents $N = [n]$, a set of goods $M = [m]$, and a boolean circuit C with m input gates. The circuit C defines a valuation function $v: 2^M \rightarrow \mathbb{N}$ which is the common valuation of all the agents. A solution is one of the following:

1. An allocation (X_1, \dots, X_n) that is EFX.
2. A pair of bundles $S \subseteq T$ that violate monotonicity, that is, $v(S) > v(T)$.

The reason for allowing the violation-of-monotonicity solutions is that the circuit C is not guaranteed to define a monotone valuation, and in this case an EFX allocation is not guaranteed to exist. Importantly, we note that our PLS-hardness result (presented in the next section) does not rely on violation solutions. In other words, even the version of the problem where we are promised that the valuation function is monotone remains PLS-hard.

Theorem 4.1. *The IDENTICAL-EFX problem lies in PLS.*

The problem of computing an EFX allocation for two non-identical agents with valuations v_1 and v_2 is reducible to the problem of computing an EFX allocation for two identical agents via the cut-and-choose protocol. As a result, we immediately also obtain the following:

Corollary 2. *Computing an EFX allocation for two not necessarily identical agents is in PLS.*

Proof. To show that the IDENTICAL-EFX problem is in PLS, we reduce it to a basic PLS problem. An instance of this basic PLS problem is just an instance of the IDENTICAL-EFX problem, i.e., a tuple $I = (N, M, C)$. The set of feasible solutions F_I is the set of all possible allocations of the goods in M to the agents in N . As an initial feasible solution, we simply take the allocation where one agent receives every good. It remains to specify the objective function c_I and the neighborhood structure N_I , and then to argue that a local optimum corresponds to an EFX allocation.

Plaut and Roughgarden [2020, Section 4] introduce the *leximin++* ordering on the set of allocations, and show that the maximum element with respect to that ordering must be an EFX allocation. In fact, a closer inspection of their proof reveals that even a *local* maximum with respect to the *leximin++* ordering must be an EFX allocation. As a result, we construct an objective function that implements the *leximin++* ordering and then use the same arguments as Plaut and Roughgarden [2020, Theorem 4.2].

For an allocation (X_1, \dots, X_n) , we let $O^X = (i_1, \dots, i_n)$ be an ordering of the agents according to increasing values of $v(X_i)$ (if multiple agents have bundles of equal utility, we break ties by ordering tied agents in terms of their agent number, i.e., if agents i and j are tied, and $i < j$, then agent i will appear before agent j in the ordering). The objective value is then defined as

$$\begin{aligned} c_I(X) = & |X_{i_n}| + v(X_{i_n})K \\ & + |X_{i_{n-1}}|K^2 + v(X_{i_{n-1}})K^3 \\ & + \dots \\ & + |X_{i_1}|K^{2n-2} + v(X_{i_1})K^{2n-1} \end{aligned}$$

where K is an upper bound on the size or utility of any bundle. We claim that if an allocation X is not EFX, then one may construct an allocation X' from X by moving a single good from one bundle to another such that the objective strictly increases, $c_I(X') > c_I(X)$. Therefore, we will consider local maximization of this objective and we define the neighborhood of X to be $N_I(X) = \{X' \in F_I : \exists i, j \in N, \exists g \in X_j \text{ s.t. } X'_i = X_i \cup \{g\}, X'_j = X_j \setminus \{g\}, X'_k = X_k \text{ for } k \neq i, j\}$. We note that the cardinality of $N_I(X)$ is polynomial in n and m , so the algorithm for finding an improving neighbor if one exists may simply compute the objective value for every allocation in the neighborhood. Thus, this local maximization problem is indeed a basic PLS problem.

Finally, we have to show that any local maximum $X \in F_I$ yields a solution to the IDENTICAL-EFX problem, i.e., X is an EFX allocation or X yields a violation of monotonicity. We say that an allocation X yields a violation of monotonicity, if there exist $i \in N$ and $g \in M$ such that $v(X_i \setminus \{g\}) > v(X_i)$ or $v(X_i \cup \{g\}) < v(X_i)$. We note that if X yields a violation of monotonicity, then the violation can be found in polynomial time.

Consider an allocation $X \in F_I$ that is not EFX and that does not yield a violation of monotonicity. We will show that X cannot be a local maximum, which then implies the desired statement by contrapositive. Since X is not EFX, we may find $i, j \in N$ and $g \in X_j$ such that $v(X_i) < v(X_j \setminus \{g\})$. Without loss of generality, we may assume that $i = \arg \min_{k \in N} v(X_k)$, and if more than one agent attains this minimum, then we take the i that appears last among those tied agents in O^X according to the tie-breaking. Now define an allocation X' by

$$\begin{aligned} X'_i &= X_i \cup \{g\} \\ X'_j &= X_j \setminus \{g\} \\ X'_k &= X_k \text{ for } k \neq i, j \end{aligned}$$

and note that $X' \in N_I(X)$. We claim that $c_I(X') > c_I(X)$, meaning that X is not a local maximum.

In order to prove this, we first show that the orderings O^X and $O^{X'}$ agree in their first ℓ positions, where $\ell \in \{0, 1, \dots, n-1\}$ is the index such that $O_{\ell+1}^X = i$, i.e., agent i appears in

position $\ell + 1$ in O^X . Let S denote the set of agents that appear in O^X before agent i , i.e., the first ℓ agents appearing in O^X . Note that S consists of all the agents that have utility $v(X_i)$ in allocation X , excluding i . First, observe that $j \notin S$, because $v(X_j) \geq v(X_j \setminus \{g\}) > v(X_i)$ as X does not yield a violation of monotonicity. Therefore, we find that the bundles of the agents in S are not changed from allocation X to X' , and, in particular, these agents still have utility $v(X_i)$ in allocation X' . Furthermore, in allocation X' , all other agents (except possibly i) have strictly larger utility than S -agents, namely $v(X'_j) = v(X_j \setminus \{g\}) > v(X_i)$, and $v(X'_k) = v(X_k) > v(X_i)$ for $k \notin S \cup \{i, j\}$. Finally, $v(X'_i) = v(X_i \cup \{g\}) \geq v(X_i)$ as X does not yield a violation of monotonicity, and thus, in allocation X' , agent i is either also tied with the agents in S , or it has strictly larger utility. In any case, by the tie-breaking, the first ℓ positions of O^X and $O^{X'}$ are the same.

We now argue that $c_I(X') > c_I(X)$. Since O^X and $O^{X'}$ agree in their first ℓ positions, and the bundles of those first ℓ agents have not changed, the 2ℓ highest-order terms in $c_I(X)$ and $c_I(X')$ have identical coefficients. By definition, $O_{\ell+1}^X = i$. If $O_{\ell+1}^{X'} = i$, then we have that $c_I(X') > c_I(X)$, because $v(X'_i) = v(X_i \cup \{g\}) \geq v(X_i)$ and $|X'_i| = |X_i \cup \{g\}| > |X_i|$, meaning that the coefficient in front of $K^{2n-(2\ell+1)}$ is at least as large in $c_I(X')$ as in $c_I(X)$ and the coefficient in front of $K^{2n-(2\ell+2)}$ is strictly larger. If $O_{\ell+1}^{X'} = k \neq i$, then we have that $c_I(X') > c_I(X)$, because $v(X'_k) > v(X_i)$, implying that the coefficient in front of $K^{2n-(2\ell+1)}$ is strictly larger in $c_I(X')$ than in $c_I(X)$. We conclude that X is not a local maximum. Therefore, by contraposition, a local maximum is an EFX allocation or it yields a violation of monotonicity. \square

4.2 PLS-hardness

In this section we prove the following theorem.

Theorem 4.2. *The problem of computing an EFX allocation for two identical agents with a submodular valuation function is PLS-hard.*

The reduction consists of two steps. First, following [Plaut and Roughgarden \[2020\]](#), we reduce the problem of local optimization on an odd Kneser graph to the problem of computing an EFX allocation for two agents sharing the same submodular valuation function. Then, in the second step, which is also our main technical contribution, we show that the PLS-complete problem FLIP reduces to local optimization on an odd Kneser graph.

4.2.1 KNESER \leq IDENTICAL-EFX

For $k \in \mathbb{N}$, the odd Kneser graph $K(2k+1, k)$ is defined as follows: the vertex set consists of all subsets of $[2k+1]$ of size k , and there is an edge between two vertices if the corresponding sets are disjoint. We identify the vertex set of $K(2k+1, k)$ with the set $\{x \in \{0, 1\}^{2k+1} : \|x\|_1 = k\}$, where $\|x\|_1 = \sum_{i=1}^{2k+1} x_i$ denotes the 1-norm. Note that there is an edge between x and x' if and only if $\langle x, x' \rangle = 0$, where $\langle \cdot, \cdot \rangle$ denotes the inner product.

Definition 4.4 (KNESER). The KNESER problem of local optimization on an odd Kneser graph is defined as the following basic PLS problem. An instance of the KNESER problem consists of a boolean circuit C with $2k+1$ input nodes for some $k \in \mathbb{N}$. The set of feasible solutions is $F_C = \{x \in \{0, 1\}^{2k+1} : \|x\|_1 = k\}$, and the neighborhood of some $x \in F_C$ is given by $N_C(x) = \{x' \in F_C : \langle x, x' \rangle = 0\}$. The goal is to find a solution that is a local maximum with respect to the objective function $C(x) = \sum_{i=0}^{m-1} y_i \cdot 2^i$, where y_0, \dots, y_{m-1} denote the output nodes of the circuit C .

Lemma 4. *KNESER reduces to IDENTICAL-EFX with two identical submodular agents.*

Proof. Our proof of this lemma closely follows the corresponding proof of [Plaut and Roughgarden \[2020, Theorem 3.1\]](#), with some minor modifications due to the different computational model.

First, we describe the map f taking instances C of KNESER to instances of IDENTICAL-EFX. We consider a valuation on subsets of $[2k + 1]$ given by

$$v(X) = \begin{cases} 2|X| & \text{if } |X| < k \\ 2k - 2^{-C(X)} & \text{if } |X| = k \\ 2k & \text{if } |X| > k \end{cases}$$

Using the description of the circuit C , we may in polynomial time construct a boolean circuit computing v . This valuation may take non-integer values, but this can be fixed by scaling by a larger power of 2. Scaling will not change anything in the arguments below. We now define $f(C) = ([2], [2k + 1], v)$. That is, the KNESER instance C is mapped to an IDENTICAL-EFX instance with $2k + 1$ goods and with two agents sharing the same valuation v .

We note that $2^{-C(X)} \in (0, 1]$, because C takes values in the natural numbers. This ensures that the valuation v is monotone, because $v(S)$ is seen to be non-decreasing in $|S|$. Therefore, the only optimal solutions of $f(C)$ are EFX allocations (X_1, X_2) . Note by inspection of v that if (X_1, X_2) is EFX, then $|X_1| = k$ and $|X_2| = k + 1$ (or $|X_1| = k + 1$ and $|X_2| = k$). If we are in the first case then X_1 corresponds to a feasible solution of the KNESER instance C . Also any neighbor of X_1 in the Kneser graph is of the form $X_2 \setminus \{g\}$ for some $g \in X_2$. As (X_1, X_2) is EFX we have that

$$\begin{aligned} 2k - 2^{-C(X_1)} &= v(X_1) \\ &\geq v(X_2 \setminus \{g\}) = 2k - 2^{-C(X_2 \setminus \{g\})} \end{aligned}$$

implying that $C(X_1) \geq C(X_2 \setminus \{g\})$ for all $g \in X_2$. We conclude that X_1 is a local maximum for the instance of KNESER given by the circuit C . Similarly, when $|X_2| = k$, X_2 will be a local maximum. As a result, we can define the polynomial-time map g that maps solutions of the IDENTICAL-EFX instance to solutions of the KNESER-instance by

$$g((X_1, X_2), C) = \begin{cases} X_1 & \text{if } |X_1| = k \\ X_2 & \text{otherwise} \end{cases}$$

By the discussion above it follows that if (X_1, X_2) is a solution to the IDENTICAL-EFX instance, then $g((X_1, X_2), C)$ is an optimal solution to the KNESER-instance. Therefore, the pair (f, g) constitutes a reduction from KNESER to IDENTICAL-EFX.

Finally, we show that v is submodular. For any $X \subseteq [2k + 1]$ and $x \notin X$ we have that

$$v(X \cup \{x\}) - v(X) = \begin{cases} 2 & \text{if } |X| < k - 1 \\ 2 - 2^{-C(X \cup \{x\})} & \text{if } |X| = k - 1 \\ 2^{-C(X)} & \text{if } |X| = k \\ 0 & \text{if } |X| > k \end{cases}$$

Using that $2^{-C(X)} \in (0, 1]$, this shows that $v(X \cup \{x\}) - v(X)$ is non-increasing in $|X|$. Thus, if $Y \subseteq X$ and $x \notin X$, we have that $v(X \cup \{x\}) - v(X) \leq v(Y \cup \{x\}) - v(Y)$, meaning that v is submodular. \square

4.2.2 FLIP \leq KNESER

Johnson et al. [1988] introduced the computational problem FLIP and proved that it is PLS-complete. We will now reduce from FLIP to KNESER to show that KNESER, and thus IDENTICAL-EFX, are PLS-hard. In particular, this also establishes the PLS-completeness of KNESER, which might be of independent interest.

Definition 4.5 (FLIP). The FLIP problem is the following basic PLS problem. The instances of FLIP are boolean circuits. For an instance C with n input nodes x_0, \dots, x_{n-1} and m output nodes y_0, \dots, y_{m-1} , the set of feasible solutions is all the possible inputs to the circuit: $F_C = \{0, 1\}^n$. For any $x \in \{0, 1\}^n$, the neighborhood is all the inputs that can be obtained from x by flipping one bit: $N_C(x) = \{x' \in \{0, 1\}^n : \Delta(x, x') = 1\}$ where $\Delta(\cdot, \cdot)$ denotes the Hamming distance. The goal is to find a solution that is locally minimal with respect to the objective function defined by $C(x) = \sum_{i=0}^{m-1} y_i \cdot 2^i$.

Lemma 5. FLIP reduces to KNESER.

Proof. We construct a reduction from FLIP to the minimization version of KNESER. The minimization version of KNESER is seen to be equivalent to its maximization version by negating the output nodes of the original circuit. Let C_F be an instance of FLIP. Denote by $p = \text{poly}(|C_F|)$ the length of the feasible solutions of C_F . The map of instances f now takes C_F to an instance C_K of the KNESER-problem whose feasible solutions are $F_K = \{x \in \{0, 1\}^{2p+1} : \|x\|_1 = p\}$. A typical feasible solution will be written as $s = uvb$ where $u, v \in \{0, 1\}^p$ and $b \in \{0, 1\}$. We will use the notation \bar{u} to denote the bitwise negation of $u \in \{0, 1\}^p$. The circuit C_K is defined as follows:

1. $C_K(u\bar{u}0) = 2 \cdot C_F(u)$,
2. $C_K(uv1) = 2 \cdot \min(C_F(\bar{u}), C_F(v)) + 1$ if $\Delta(\bar{u}, v) = 1$,
3. $C_K(uvb) = M + \Delta(\bar{u}, v)$ otherwise.

Here M denotes a number chosen to be sufficiently large so that it dominates any cost $2 \cdot C_F(w)$. Note that the circuit C_K is well-defined and that it can be constructed in polynomial time given the circuit C_F . At a high level, the definition of the cost of a vertex of the third type is meant to ensure that for any such vertex uvb , there is a sequence of neighbors with decreasing costs that ends in a vertex of the form $u\bar{u}0$. The costs of the first and second vertex types are then meant to ensure that for a vertex $u\bar{u}0$, there is a sequence of neighbors with decreasing costs that ends in a vertex $w\bar{w}0$ where w is an improving neighbor of u in the original FLIP-instance.

Below we show that the only local minima of C_K are of the form $u\bar{u}0$ where u is a local minimum for C_F . Therefore, upon defining the solution-mapping by $g(uvb) = u$ we have that (f, g) is a reduction from FLIP to KNESER.

No optimal solutions of type (3). If a feasible solution $s = uvb$ is of type (3), then we claim that it must have a neighbor of lower cost. First of all, note that since s is not of type (1) or (2), and since $\|s\|_1 = p$, it follows that $\Delta(\bar{u}, v) \geq 2$. Now, because $\Delta(\bar{u}, v) \geq 2 > 0$ and $\|uv\|_1 \leq p$, there must exist an i such that $u_i = v_i = 0$. Otherwise one would find that $\|uv\|_1 > p$, which contradicts s being a feasible solution. Now, let $s' = u'v'b'$, where $u' = \bar{u}$, $b' = \bar{b}$, and $v'_j = \bar{v}_j$ for all $j \neq i$, but $v'_i = v_i = 0$. We note that $\|s'\|_1 = \|\bar{s}\|_1 - 1 = (p+1) - 1 = p$, so s' is a valid vertex in the Kneser graph. Further, we see that s' is a neighbor of s , because $s'_j s_j = 0$ for all j . If s' is not of type (3), then it has lower cost than s by construction of C_K and choice of M . Finally, if s' is of type (3), then the observation that $\Delta(\bar{u}', v') < \Delta(\bar{u}, v)$ again yields that s' has lower cost than s .

No optimal solutions of type (2). Suppose $s = uv1$ is of type (2). As $\|s\|_1 = p$ and $\Delta(\bar{u}, v) = 1$, there is some i with $v_i = 0$ and $\bar{u}_i = 1$, and $v_j = \bar{u}_j$ for $j \neq i$. This implies that $\sum_i u_i v_i = 0$, and so both $s' = \bar{u}u0$ and $s'' = v\bar{v}0$ are neighbors of s . Furthermore, by construction of C_K , the cost of s' or of s'' is strictly less than the cost of s .

Optimal solutions. Consider a feasible solution of the form $u\bar{u}0$. If u is not a local minimum for C_F , then let w be an improving neighbor of u . As $\Delta(u, w) = 1$, there are now two cases to consider. If $u_i = 0$ and $w_i = 1$ for some i , then $s' = \bar{w}u1$ is a type (2) neighbor of lower cost. If $u_i = 1$ and $w_i = 0$ for some i , then $s' = \bar{u}w1$ is a type (2) neighbor of lower cost. Therefore, if $u\bar{u}0$ is a local minimum for C_K , then u is a local minimum for C_F . \square

Corollary 3. *Let $n \geq 2$ be an integer. Computing an EFX allocation for n identical agents with a submodular valuation function is PLS-hard.*

Proof. By Theorem 4.2 it suffices to produce a reduction from the problem of computing an EFX allocation for two identical agents to the problem of computing an EFX allocation for n identical agents. We sketch this reduction. Let $u: 2^M \rightarrow \mathbb{R}$ denote the common submodular valuation function of the two agents. Construct an EFX-instance with n agents by adding $n - 2$ agents and $n - 2$ goods, $M' = M \cup \{g_1, \dots, g_{n-2}\}$. Define the valuation function of the n agents to be $u' = \bar{u} + v$ where $\bar{u}: 2^{M'} \rightarrow \mathbb{R}$ is the extension of u given by $\bar{u}(S) = u(S \cap M)$ and where $v: 2^{M'} \rightarrow \mathbb{R}$ is additive given by $v(\{g_i\}) = u(M) + 1$ for $i = 1, \dots, n - 2$ and $v(\{g\}) = 0$ for $g \in M$. One may verify that \bar{u} is submodular, and so that u' is the sum of two submodular valuations and therefore itself submodular.

Let (X_1, \dots, X_n) denote an EFX allocation of this instance. We claim that after permuting the bundles, we may assume that $X_{i+2} = \{g_i\}$ for $i = 1, \dots, n - 2$ and $X_1 \cup X_2 = M$. At least one bundle, say X_1 , receives no good from $\{g_1, \dots, g_{n-2}\}$, and so $u'(X_1) = u(X_1) \leq u(M)$. Now suppose some other bundle X_i contains some good g_j . If X_i contained another good g , then

$$u'(X_i \setminus \{g\}) \geq u'(\{g_j\}) = u(M) + 1 > u'(X_1),$$

contradicting (X_1, \dots, X_n) being EFX. Hence, $X_i = \{g_j\}$, and the claim follows. Now, one sees that (X_1, X_2) is an EFX allocation of the original two-agent instance. \square

Acknowledgements

P. W. Goldberg was supported by a JP Morgan faculty award. K. Høgh was supported by the Independent Research Fund Denmark under grant no. 9040-00433B. Most of this work was done while he was visiting Oxford thanks to a STIBO IT Travel Grant. A. Hollender was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00026.

References

- Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A. Voudouris. Maximum Nash welfare and other stories about EFX. *Theoretical Computer Science*, 863:69–85, 2021. doi:[10.1016/j.tcs.2021.02.020](https://doi.org/10.1016/j.tcs.2021.02.020).
- Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris. Fair division of indivisible goods: A survey. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5385–5393, 2022. doi:[10.24963/ijcai.2022/756](https://doi.org/10.24963/ijcai.2022/756).
- Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Proceedings of the 57th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 416–427, 2016. doi:[10.1109/focs.2016.52](https://doi.org/10.1109/focs.2016.52).
- Haris Aziz, Bo Li, Herve Moulin, and Xiaowei Wu. Algorithmic fair allocation of indivisible items: A survey and new questions. *arXiv:2202.08713*, 2022.
- Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5119–5126, 2021. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16647>.
- Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 557–574, 2018. doi:[10.1145/3219166.3219176](https://doi.org/10.1145/3219166.3219176).

- Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. Almost full EFX exists for four agents. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 4826–4833, 2022. doi:[10.1609/aaai.v36i5.20410](https://doi.org/10.1609/aaai.v36i5.20410).
- Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011. doi:[10.1086/664613](https://doi.org/10.1086/664613).
- Ioannis Caragiannis, Nick Gravin, and Xin Huang. Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 527–545, 2019a. doi:[10.1145/3328526.3329574](https://doi.org/10.1145/3328526.3329574).
- 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, 2019b. doi:[10.1145/3355902](https://doi.org/10.1145/3355902).
- Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. EFX exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 1–19, 2020. doi:[10.1145/3391403.3399511](https://doi.org/10.1145/3391403.3399511).
- Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, Ruta Mehta, and Pranabendu Misra. Improving EFX guarantees through rainbow cycle number. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 310–311, 2021a. doi:[10.1145/3465456.3467605](https://doi.org/10.1145/3465456.3467605).
- Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. *SIAM Journal on Computing*, 50(4):1336–1358, 2021b. doi:[10.1137/20m1359134](https://doi.org/10.1137/20m1359134).
- Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971. doi:[10.1007/bf01584082](https://doi.org/10.1007/bf01584082).
- Duncan K. Foley. *Resource Allocation and the Public Sector*. PhD thesis, Yale University, 1966.
- David Gale. Optimal assignments in an ordered set: An application of matroid theory. *Journal of Combinatorial Theory*, 4(2):176–180, 1968. doi:[10.1016/s0021-9800\(68\)80039-0](https://doi.org/10.1016/s0021-9800(68)80039-0).
- George Gamow and Marvin Stern. *Puzzle-math*. Viking Press, 1958.
- Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. Near fairness in matroids. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, pages 393–398, 2014. doi:[10.3233/978-1-61499-419-0-393](https://doi.org/10.3233/978-1-61499-419-0-393).
- Faruk Gul and Ennio Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999. URL <https://EconPapers.repec.org/RePEc:eee:jetheo:v:87:y:1999:i:1:p:95-124>.
- David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. doi:[10.1016/0022-0000\(88\)90046-3](https://doi.org/10.1016/0022-0000(88)90046-3).
- Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006. doi:[10.1016/j.geb.2005.02.006](https://doi.org/10.1016/j.geb.2005.02.006).
- Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004. doi:[10.1145/988772.988792](https://doi.org/10.1145/988772.988792).

- Pasin Manurangsi and Warut Suksompong. Closing gaps in asymptotic fair division. *SIAM Journal on Discrete Mathematics*, 35(2):668–706, 2021. doi:[10.1137/20m1353381](https://doi.org/10.1137/20m1353381).
- Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991. doi:[10.1016/0304-3975\(91\)90200-L](https://doi.org/10.1016/0304-3975(91)90200-L).
- Renato Paes Leme. Gross substitutability: An algorithmic survey. *Games and Economic Behavior*, 106:294–316, 2017. doi:[10.1016/j.geb.2017.10.016](https://doi.org/10.1016/j.geb.2017.10.016).
- Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2):1039–1068, 2020. doi:[10.1137/19m124397x](https://doi.org/10.1137/19m124397x).
- Richard Rado. Note on independence functions. *Proceedings of the London Mathematical Society*, s3-7(1):300–320, 1957. doi:[10.1112/plms/s3-7.1.300](https://doi.org/10.1112/plms/s3-7.1.300).
- Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948. URL <https://www.jstor.org/stable/1914289>.
- Hugo Steinhaus. Sur la division pragmatique. *Econometrica*, 17(Suppl.):315–319, 1949. doi:[10.2307/1907319](https://doi.org/10.2307/1907319).
- Walter Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980. doi:[10.1080/00029890.1980.11995109](https://doi.org/10.1080/00029890.1980.11995109).
- Francis Edward Su. Rental harmony: Sperner’s lemma in fair division. *The American Mathematical Monthly*, 106(10):930–942, 1999. doi:[10.1080/00029890.1999.12005142](https://doi.org/10.1080/00029890.1999.12005142).
- Hal R. Varian. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1):63–91, 1974. doi:[10.1016/0022-0531\(74\)90075-1](https://doi.org/10.1016/0022-0531(74)90075-1).
- Douglas R. Woodall. Dividing a cake fairly. *Journal of Mathematical Analysis and Applications*, 78(1):233–247, 1980. doi:[10.1016/0022-247x\(80\)90225-5](https://doi.org/10.1016/0022-247x(80)90225-5).