

Reforming an Unfair Allocation by Exchanging Goods

Sheung Man Yuen 

National University of Singapore, Singapore

Ayumi Igarashi 

University of Tokyo, Japan

Naoyuki Kamiyama 

Kyushu University, Fukuoka, Japan

Warut Suksompong 

National University of Singapore, Singapore

Abstract

Fairly allocating indivisible goods is a frequently occurring task in everyday life. Given an initial allocation of the goods, we consider the problem of reforming it via a sequence of exchanges to attain fairness in the form of envy-freeness up to one good (EF1). We present a vast array of results on the complexity of determining whether it is possible to reach an EF1 allocation from the initial allocation and, if so, the minimum number of exchanges required. In particular, we uncover several distinctions based on the number of agents involved and their utility functions. Furthermore, we derive essentially tight bounds on the worst-case number of exchanges needed to achieve EF1.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory; Theory of computation → Design and analysis of algorithms; Theory of computation → Problems, reductions and completeness

Keywords and phrases fair division, indivisible goods, envy-freeness, exchanges

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.54

Related Version *Full Version:* <https://arxiv.org/pdf/2412.19264> [34]

Funding This work was partially supported by the Singapore Ministry of Education under grant number MOE-T2EP20221-0001, by JST PRESTO under grant number JPMJPR20C1, by JSPS KAKENHI under grant number JP20H05795, by JST ERATO under grant number JPMJER2301, and by an NUS Start-up Grant.

Acknowledgements We would like to thank the anonymous reviewers for their valuable comments.

1 Introduction

Fair division is a research area that studies how to allocate scarce resources to interested agents in a fair manner. Over the past several decades, a large body of work has developed on concepts and algorithms for fairly allocating various types of resources [6, 27, 28, 30]. The developed theory has also been applied to many allocation scenarios in practice [9, 17, 19, 23].

A ubiquitous fair division problem is the allocation of *indivisible goods*, such as books, furniture, paintings, and human resources. While numerous fairness notions have been proposed for allocating indivisible goods, one of the most prominent notions is *envy-freeness up to one good (EF1)*. This notion requires that if an agent envies another agent, there must exist a good in the latter agent’s bundle whose removal would make the envy disappear. Besides its simplicity, a salient feature of EF1 is that an allocation satisfying it always exists and can be found, e.g., by the *round-robin algorithm*, which lets the agents pick their favorite goods in a round-robin fashion. The allocation returned by this algorithm is also *balanced*, meaning that the numbers of goods that any pair of agents receive differ by at most one.



© Sheung Man Yuen, Ayumi Igarashi, Naoyuki Kamiyama, and Warut Suksompong;
licensed under Creative Commons License CC-BY 4.0

36th International Symposium on Algorithms and Computation (ISAAC 2025).

Editors: Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai; Article No. 54; pp. 54:1–54:21

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The fair division literature typically assumes that there is a set of unallocated goods and the objective is to allocate them fairly. In certain scenarios, however, an existing (possibly unfair) allocation of the goods is already in place, and the goal is to “reform” it in order to arrive at a fair allocation. This is the case, for instance, in the allocation of personnel among teams in an organization. As the personnel experience individual growth or decline, and as the needs of the teams evolve, these changes can necessitate a reevaluation and potential reformation of the current allocation by the organization leadership. Another example is the distribution of a museum’s exhibits across its branches – the museum director may decide to adjust the distribution so as to ensure fairness based on the most recent valuations. Such scenarios fall under the umbrella of *control problems*, which assume a central authority with the power to control the outcome and have been studied extensively in fair division and computational social choice [11, 15].

In this work, we shall allow agents to *exchange* pairs of goods in the reformation process, and use EF1 as our fairness criterion. Exchanges constitute a fundamental type of operation and preserve the size of each agent’s bundle, thereby ensuring that any cardinality constraints remain fulfilled.¹ Naturally, given an initial allocation, we wish to reach an EF1 allocation using a small number of exchanges. However, it is sometimes impossible to reach an EF1 allocation via *any* finite number of exchanges (e.g., if one agent receives many more goods than another agent in the initial allocation), so we start by exploring whether the corresponding decision problem can be answered efficiently. Since this problem is equivalent to determining whether an EF1 allocation with a certain size vector exists in a given instance, it is also meaningful independently of exchange considerations.² We also investigate other important questions in this setting. Namely, if it is possible to reach an EF1 allocation, can we efficiently determine the smallest number of exchanges required to achieve this goal? And how many exchanges might we need to make in the worst case in order to attain EF1?

1.1 Our Results

In our model, there is an initial allocation of a set of goods. As is often assumed in fair division, each agent has an additive utility function over the goods. At each step of the reformation process, two agents can exchange a pair of goods with each other to obtain another allocation, and the goal is to reach an EF1 allocation at the end of the process. More details on our model are provided in Section 2.

In Section 3, we consider the decision problem of determining whether a given initial allocation can be reformed into an EF1 allocation. As mentioned earlier, this problem is equivalent to determining whether an EF1 allocation with a given size vector exists, so we focus on the latter decision problem instead. We demonstrate interesting distinctions in the complexity based on the number of agents and their utility functions. Specifically, in the case of two agents, the problem can be solved in polynomial time if the agents have identical utilities, but becomes (weakly) NP-complete otherwise. For three or more agents, the problem is NP-complete even with identical utilities; however, it can be solved efficiently when the utilities are binary³ provided that the number of agents is constant. Finally, for

¹ Capacity constraints are prevalent in fair division applications and have accordingly received interest in the literature [4, 21, 32, 33].

² A *size vector* specifies the number of goods that each agent receives in an allocation. When an EF1 allocation is not guaranteed to exist in some instances due to cardinality requirements, an approach taken by previous work is to relax the EF1 condition (e.g., [33]). However, this leads to unnecessarily weak guarantees in instances where EF1 can be attained.

³ This means that each agent has utility 0 or 1 for each good.

■ **Table 1** Computational complexity of REFORMABILITY, the problem of deciding whether an EF1 allocation with a given size vector exists in a given instance. The top row represents the class of utility functions considered. The leftmost column represents the number of agents. “sNP-c” and “wNP-c” stand for strongly NP-complete and weakly NP-complete respectively.

utilities	general	identical	binary	identical & binary
$n = 2$	wNP-c (Th. 6)	P (Th. 5)	P (Th. 10)	P (Th. 10)
constant $n \geq 3$	wNP-c (Th. 8)	wNP-c (Th. 8)	P (Th. 10)	P (Th. 10)
general n	sNP-c (Th. 11)	sNP-c (Th. 11)	NP-c (Th. 12)	P (Th. 13)

an arbitrary (non-constant) number of agents, the problem is strongly NP-hard even for identical utilities and NP-hard for binary utilities,⁴ but can be solved in polynomial time if the utilities are identical *and* binary. The results of this section are summarized in Table 1.

Having determined the reformability of the initial allocation, we next explore in Section 4 the problem of deciding whether the optimal (i.e., minimum) number of exchanges required to reach an EF1 allocation is at most some given number k . For (a) two agents with identical utilities, (b) a constant number of agents with binary utilities, and (c) any number of agents with identical binary utilities, we show that this problem can be solved in polynomial time, just like the corresponding decision problem in Section 3. For the remaining scenarios, since deciding whether an allocation is reformable is already NP-complete (from Section 3), we instead focus on the special case where the allocation is balanced – an EF1 allocation is guaranteed to be reachable in this case (see Proposition 2). We show that the problem for this special case remains NP-complete.

Finally, in Section 5, instead of considering specific instances, we derive worst-case bounds on the number of exchanges necessary to reach an EF1 allocation. We assume that each of the n agents possesses s goods – this again ensures that an EF1 allocation is reachable. We show that roughly $s(n-1)/2$ exchanges always suffice for instances with general utilities or with binary utilities; moreover, our bound is essentially tight for any n and s , and exactly tight when $n = 2$ as well as when s is divisible by n . For instances with identical binary utilities, we show that an essentially tight bound for the number of exchanges is $sn/4$ for even n and $s(n-1)(n+1)/4n$ for odd n .

All omitted proofs can be found in the full version of our paper [34].

1.2 Additional Related Work

As mentioned earlier, the majority of work in fair division assumes that there is no initial allocation of the resources – we now discuss the key exceptions and their differences from our model. Boehmer et al. [5] studied the problem of *discarding* goods from an initial allocation in order to reach an envy-free or EF1 allocation. As it is possible to reallocate the goods in several practical situations, discarding them can be unnecessarily wasteful for the agents involved. In a similar vein, Dorn et al. [14] investigated deleting goods to attain another fairness notion called *proportionality*; they assumed that agents have ordinal preferences (rather than cardinal utilities) over the goods, and considered both the settings with and without an initial allocation.⁵ Instead of deleting goods, Bredereck et al. [8] allowed

⁴ For binary utilities, there is no distinction between weak and strong NP-hardness.

⁵ When there is no initial allocation, Dorn et al. [14] considered deleting goods so that a proportional allocation of the remaining goods exists. Aziz et al. [2] examined discarding or adding goods to achieve envy-freeness, also in the absence of an initial allocation and under ordinal preferences.

agents to share goods in order to improve their allocations, while Bentert et al. [3] analyzed the resolution of envy by adding goods. Aziz et al. [1] focused on reallocating goods to make agents better off, but did not delve into the aspect of fairness. Igarashi et al. [22] aimed to transition from an initial allocation to a target allocation, both of which are EF1, while maintaining EF1 throughout the process. Chandramouleeswaran et al. [10] examined *transferring* goods starting from a “near-EF1” allocation with the goal of reaching an EF1 allocation. Segal-Halevi [31] considered the reallocation of a *divisible* good and explored the trade-off between guaranteeing a minimum utility for every agent and ensuring each agent a certain fraction of her original utility. Chevaleyre et al. [12] also strived to reach fair allocations but via exchanges with money.

Further afield, the idea of improving an initial allocation has also been examined when each agent receives only one good, a setting sometimes known as a *housing market*. Gourvès et al. [18] assumed an underlying social network and allowed beneficial exchanges between agents who are neighbors in the network – their work led to a series of follow-up papers on similar models [20, 25, 26, 29]. Damamme et al. [13] also considered exchanges but without an underlying graph structure, while Brandt and Wilczynski [7] used *Pareto-optimality* as their target notion. The papers mentioned so far in this paragraph did not have fairness as their objective. Ito et al. [24] incorporated fairness in the form of envy-freeness into this setting – starting with an envy-free allocation, they let each agent exchange her current good with a preferred unassigned good as long as the exchange keeps the allocation envy-free.

2 Preliminaries

Let $N = \{1, \dots, n\}$ be a set of $n \geq 2$ agents, and M be a set of $m \geq 1$ goods typically denoted by $g_1, \dots, g_m$. A *bundle* is a (possibly empty) subset of goods. An *allocation* $\mathcal{A} = (A_1, \dots, A_n)$ is an ordered partition of M into n bundles such that bundle A_i is allocated to agent $i \in N$. An (*allocation*) *size vector* $\vec{s} = (s_1, \dots, s_n)$ is a vector of non-negative integers such that $\sum_{i \in N} s_i = m$. We say that an allocation $\mathcal{A}$ has size vector $\vec{s}$ if $|A_i| = s_i$ for all $i \in N$. A size vector $\vec{s}$ is *balanced* if $|s_i - s_j| \leq 1$ for all $i, j \in N$, and an allocation is *balanced* if it has a balanced size vector.

Each agent $i \in N$ has an additive *utility function* $u_i : 2^M \rightarrow \mathbb{R}_{\geq 0}$ that maps bundles to non-negative real numbers; additivity means that $u_i(M') = \sum_{g \in M'} u_i(\{g\})$ for all $i \in N$ and $M' \subseteq M$. We write $u_i(g)$ instead of $u_i(\{g\})$ for a single good $g \in M$. The utility functions are *identical* if $u_i = u_j$ for all $i, j \in N$ – we shall use u to denote the common utility function in this case. The utility functions are *binary* if $u_i(g) \in \{0, 1\}$ for all $i \in N$ and $g \in M$. Agent i is *EF1 towards* agent j in an allocation $\mathcal{A} = (A_1, \dots, A_n)$ if either $A_j = \emptyset$ or there exists a good $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{g\})$. An allocation $\mathcal{A}$ is *EF1* if every agent is EF1 towards every other agent in $\mathcal{A}$. A (*fair division*) *instance* $\mathcal{I}$ consists of a set of agents N , a set of goods M , and the agents’ utility functions $(u_i)_{i \in N}$.

An *exchange* involves an agent i giving one good from her bundle to agent j and simultaneously receiving one good from agent j ’s bundle. We say that allocations $\mathcal{A} = (A_1, \dots, A_n)$ and $\mathcal{B} = (B_1, \dots, B_n)$ can be *reached via an exchange* if there exist distinct $i, j \in N$, $g \in A_i$, and $g' \in A_j$ such that $B_i = (A_i \cup \{g'\}) \setminus \{g\}$, $B_j = (A_j \cup \{g\}) \setminus \{g'\}$, and $B_k = A_k$ for all $k \in N \setminus \{i, j\}$. An allocation $\mathcal{B}$ can be *reached* from an allocation $\mathcal{A}$ if there exist a non-negative integer T and a sequence of allocations $(\mathcal{A}^0, \mathcal{A}^1, \dots, \mathcal{A}^T)$ such that $\mathcal{A}^0 = \mathcal{A}$, $\mathcal{A}^T = \mathcal{B}$, and for each $t \in \{0, \dots, T-1\}$, $\mathcal{A}^t$ and $\mathcal{A}^{t+1}$ can be reached via an exchange. The *optimal number of exchanges* required to reach $\mathcal{B}$ from $\mathcal{A}$ is the smallest T

across all such sequences of allocations – if no such T exists (i.e., $\mathcal{B}$ cannot be reached from $\mathcal{A}$), then the optimal number of exchanges is defined to be ∞ .⁶ Observe that two allocations can be reached from each other if and only if they share the same size vector.

► **Proposition 1.** *Given an instance, let $\mathcal{A}$ and $\mathcal{B}$ be allocations in the instance. Then, $\mathcal{B}$ can be reached from $\mathcal{A}$ if and only if $\mathcal{A}$ and $\mathcal{B}$ have the same size vector.*

Next, we state a simple proposition that characterizes the existence of EF1 allocations based on the size vector.

► **Proposition 2.** *Let $\vec{s} = (s_1, \dots, s_n)$, and let $m = \sum_{i=1}^n s_i$.*

- (a) *If $\vec{s}$ is balanced, then every instance with n agents and m goods admits an EF1 allocation with size vector $\vec{s}$.*
- (b) *If $\vec{s}$ is not balanced, then there exists an instance with n agents and m goods that does not admit any EF1 allocation with size vector $\vec{s}$.*

Finally, we introduce an NP-hard decision problem called the BALANCED MULTI-PARTITION problem, which we will use later in the proofs of several results. In BALANCED MULTI-PARTITION, we are given positive integers p, q, K and a multiset of positive integers $X = \{x_1, \dots, x_{pq}\}$ such that $K < x_j \leq 2K$ for all $j \in \{1, \dots, pq\}$, and the sum of all the integers in X is $p(q+1)K$. The problem is to decide whether X can be partitioned into multisets $X_1, \dots, X_p$ of equal cardinalities and sums, i.e., for each $i \in \{1, \dots, p\}$, the cardinality of X_i is q and the sum of all the integers in X_i is $(q+1)K$. The NP-hardness of this problem is based on a reduction from the equal-cardinality version of the NP-hard problem PARTITION [16, p. 223].

► **Proposition 3.** *For any fixed $p \geq 2$, BALANCED MULTI-PARTITION is NP-hard.*

3 Reformability of Allocations

We start by investigating the decision problem of whether a given initial allocation can be reformed into an EF1 allocation. By Proposition 1, this reformation is possible if and only if there exists an EF1 allocation with the same size vector as the initial allocation. Therefore, in the rest of this section, we shall equivalently focus on the problem of deciding the existence of an EF1 allocation with a given size vector – this problem can be of interest independently of reformation considerations, e.g., when space constraints are present.

Now, Proposition 2 tells us that an EF1 allocation with a balanced size vector always exists. This means that the only time when we may have difficulties in ascertaining whether an EF1 allocation exists is when the given size vector is *not* balanced. In fact, as some of our proofs in this section show, the decision problem is NP-hard even when the sizes of the agents' bundles differ by *exactly two* (e.g., in Theorem 6).

We discuss the cases of two agents, a constant number of agents, and a general number of agents separately. For each case, we explore how the hardness of the decision problem varies across different classes of utility functions. Our results are summarized in Table 1.

⁶ The optimal number of exchanges can be viewed as the distance between the two allocations in the implicit *exchange graph*, where the allocations are vertices and the edges connect allocations that can be reached via an exchange [22]. When there is no path connecting two vertices of a graph, it is common to define the distance between them as infinity.

For convenience, we refer to as **REFORMABILITY** the problem of deciding whether an EF1 allocation with a given size vector exists in a given instance. Note that **REFORMABILITY** is in NP regardless of the number of agents, as we can verify in polynomial time whether a given allocation satisfies the condition by simply checking its size vector and comparing the bundles of every pair of agents for EF1.

3.1 Two Agents

For two agents, interestingly, the computational complexity of the problem turns out to be different depending on whether the agents have identical utilities or not. We begin our discussion with the case of identical utilities.

For two agents with identical utilities, we first provide a characterization for the existence of a desired EF1 allocation based on the size vector and the utilities of the goods. We show that an EF1 allocation with a given size vector exists if and only if the agent with fewer goods (say, agent 1) is EF1 towards the other agent (say, agent 2) in the allocation where agent 1 receives the most valuable goods. Note that the condition in the lemma only requires checking that agent 1 is EF1 towards agent 2; in particular, it does not require checking that agent 2 is EF1 towards agent 1.

► **Lemma 4.** *Given an instance with two agents with identical utilities, let $\vec{s} = (s_1, s_2)$ be a size vector with $s_1 \leq s_2$. Assume that the goods $g_1, \dots, g_m$ are arranged in non-increasing order of utility, and let $M_0 = \{g_1, \dots, g_{s_1}\}$. Then, there exists an EF1 allocation with size vector $\vec{s}$ if and only if agent 1 is EF1 towards agent 2 in the allocation $(M_0, M \setminus M_0)$.*

Proof. We say in this proof that for any nonempty set $M' \subseteq M$, the good $g_i \in M'$ is the *most valuable* good in M' if g_i is the good with the smallest index in M' ; likewise, g_i is the *least valuable* good in M' if g_i is the good with the largest index in M' . Note that the most (resp. least) valuable good in M' is the one with the highest (resp. lowest) utility among all the goods in M' , with ties broken by index.

($\Rightarrow$) Let (A_1, A_2) be an EF1 allocation with size vector $\vec{s}$. Let g and g' be the most valuable good in A_2 and $M \setminus M_0$ respectively. Since M_0 is the set containing the s_1 most valuable goods, we have $u(M_0) \geq u(A_1)$. Since (A_1, A_2) is an EF1 allocation, we have $u(A_1) \geq u(A_2 \setminus \{g\})$. Moreover, since $M \setminus M_0$ is the set containing the s_2 least valuable goods, we have $u(A_2 \setminus \{g\}) \geq u((M \setminus M_0) \setminus \{g'\})$. Combining the three inequalities, we get $u(M_0) \geq u((M \setminus M_0) \setminus \{g'\})$. It follows that agent 1 is EF1 towards agent 2 in the allocation $(M_0, M \setminus M_0)$.

($\Leftarrow$) Suppose that agent 1 is EF1 towards agent 2 in the allocation $(M_0, M \setminus M_0)$. If agent 2 is also EF1 towards agent 1 in $(M_0, M \setminus M_0)$, then we are done; therefore, assume that agent 2 envies agent 1 by more than one good. For notational simplicity, let $h_j = g_{s_1+j}$ for $j \in \{1, \dots, s_1\}$, so that the goods arranged in non-increasing order of utility are $g_1, g_2, \dots, g_{s_1}, h_1, h_2, \dots, h_{s_1}, g_{2s_1+1}, \dots, g_m$. Let $A_1^1 = M_0 = \{g_1, \dots, g_{s_1}\}$ and $A_2^1 = M \setminus M_0 = \{h_1, \dots, h_{s_1}\} \cup \{g_{2s_1+1}, \dots, g_m\}$.

Let $t = 1$. In the allocation (A_1^t, A_2^t) , agent 1 is EF1 towards agent 2, but agent 2 envies agent 1 by more than one good. Since g_t is the most valuable good in A_1^t , we have $u(A_2^t) < u(A_1^t \setminus \{g_t\})$. Let $A_1^{t+1} = (A_1^t \cup \{h_t\}) \setminus \{g_t\}$ and $A_2^{t+1} = (A_2^t \cup \{g_t\}) \setminus \{h_t\}$ be the bundles after exchanging g_t and h_t . Then, we have

$$\begin{aligned} u(A_1^{t+1}) &= u((A_1^t \cup \{h_t\}) \setminus \{g_t\}) \\ &\geq u(A_1^t \setminus \{g_t\}) > u(A_2^t) = u((A_2^{t+1} \cup \{h_t\}) \setminus \{g_t\}) \geq u(A_2^{t+1} \setminus \{g_t\}), \end{aligned}$$

so agent 1 is EF1 towards agent 2 in (A_1^{t+1}, A_2^{t+1}) . If agent 2 is also EF1 towards agent 1, then (A_1^{t+1}, A_2^{t+1}) is an EF1 allocation and we are done. Otherwise, agent 2 envies agent 1 by more than one good, and we increment t by 1 and repeat the discussion in this paragraph.

If we still have not found an EF1 allocation after $t = s_1$, then agent 1 is EF1 towards agent 2 in $(A_1^{s_1+1}, A_2^{s_1+1})$, where $A_1^{s_1+1} = \{h_1, \dots, h_{s_1}\} \subseteq A_2^1$ and $A_2^{s_1+1} = \{g_1, \dots, g_{s_1}\} \cup \{g_{2s_1+1}, \dots, g_m\} \supseteq A_1^1$, and g_1 is the most valuable good in $A_2^{s_1+1}$. This implies that

$$u(A_1^{s_1+1}) \leq u(A_2^1) < u(A_1^1 \setminus \{g_1\}) \leq u(A_2^{s_1+1} \setminus \{g_1\}),$$

which means that agent 1 is *not* EF1 towards agent 2 in $(A_1^{s_1+1}, A_2^{s_1+1})$. This is a contradiction; therefore, (A_1^t, A_2^t) must be EF1 for some $t \in \{1, \dots, s_1\}$. $\blacktriangleleft$

Since the condition in Lemma 4 can be checked in polynomial time, we can derive the following result.

► **Theorem 5.** *REFORMABILITY is in P for two agents with identical utilities.*

While deciding whether an EF1 allocation with a given size vector exists can be done efficiently for two agents with identical utilities, we remark here that deciding whether an *envy-free* allocation exists is NP-hard for two agents with identical utilities even if we allow any size vector – this follows directly from a reduction from PARTITION.⁷

We now proceed to general utilities. Lemma 4 assumes identical utilities, and there is no obvious way to generalize it to non-identical utilities. In fact, perhaps surprisingly, we show that the decision problem becomes NP-hard when we drop the assumption of identical utilities. The proof follows from a reduction from BALANCED MULTI-PARTITION with $p = 2$, an NP-hard problem by Proposition 3.

► **Theorem 6.** *REFORMABILITY is weakly NP-complete for two agents.*

Proof. Clearly, this problem is in NP. The “weak” aspect is demonstrated later in Lemma 7, which says that there exists a pseudopolynomial-time algorithm that solves this problem for any constant number of agents. Therefore, it suffices to show that this problem is NP-hard.

To demonstrate NP-hardness, we shall reduce from the NP-hard problem BALANCED MULTI-PARTITION with $p = 2$ (see Proposition 3). Let a BALANCED MULTI-PARTITION instance be given with $p = 2$. Without loss of generality, assume that $q \geq 2$. Let $Y = \{y_1, \dots, y_{2q+2}\}$ be a multiset such that $y_j = x_j$ for $j \in \{1, \dots, 2q\}$, $y_{2q+1} = 2K$, and $y_{2q+2} = 0$. We claim that Y can be partitioned into two multisets Y_1 and Y_2 of equal cardinalities (i.e., of size $q + 1$ each) with sums $(q + 3)K$ and $(q + 1)K$ respectively if and only if X can be partitioned into two multisets X_1 and X_2 of equal cardinalities and sums. If the latter condition holds, then let Y_1 (resp. Y_2) contain the corresponding elements in X_1 (resp. X_2), and let $y_{2q+1} \in Y_1$ and $y_{2q+2} \in Y_2$ – this gives an appropriate partition of Y . Conversely, if the former condition holds, then we show that X can be partitioned appropriately. Note that if $y_{2q+1} \in Y_2$, then there are at least $q - 1 > 0$ integers in $\{y_1, \dots, y_{2q}\}$ that are also in Y_2 . Since every integer in $\{y_1, \dots, y_{2q}\}$ is more than K , the sum of Y_2 will be more than $(q - 1)K + 2K = (q + 1)K$, which is a contradiction. This means that $y_{2q+1} \in Y_1$. Similarly, if $y_{2q+2} \in Y_1$, then there are exactly $q + 1$ integers in $\{y_1, \dots, y_{2q}\}$ that are in Y_2 . The sum of Y_2 will be more than $(q + 1)K$, which is a contradiction. Hence, $y_{2q+2} \in Y_2$. Now, this means that $\{y_1, \dots, y_{2q}\}$ must be partitioned into two multisets of equal cardinalities with sum $(q + 1)K$ each. This induces an appropriate partition of X .

⁷ If we require both agents to receive the same number of goods, the problem for envy-freeness remains NP-hard by a reduction from the equal-cardinality version of PARTITION.

Next, define a fair division instance as follows. There are $n = 2$ agents and a set of goods $M = \{g_1, \dots, g_{2q+6}\}$. Agent 2's utility is such that $u_2(g_j) = y_j$ for $j \in \{1, \dots, 2q+2\}$, $u_2(g_{2q+3}) = u_2(g_{2q+4}) = 0$, and $u_2(g_{2q+5}) = u_2(g_{2q+6}) = 2K$. Agent 1's utility is such that $u_1(g) = u_2(g) + 4K$ for all $g \in M$. The size vector $\vec{s}$ is $(q+2, q+4)$. This reduction can be done in polynomial time. We claim that there exists an EF1 allocation with size vector $\vec{s}$ in this instance if and only if Y can be partitioned into two multisets Y_1 and Y_2 of equal cardinalities (i.e., of size $q+1$ each) with sums $(q+3)K$ and $(q+1)K$ respectively.

($\Leftarrow$) Let J'_1 and J'_2 be a partition of $\{1, \dots, 2q+2\}$ of equal cardinalities such that $\sum_{j \in J'_1} y_j = (q+3)K$ and $\sum_{j \in J'_2} y_j = (q+1)K$. Let $A_1 = \{g_j \mid j \in J'_1\} \cup \{g_{2q+5}\}$ and $A_2 = M \setminus A_1$ be the two agents' bundles respectively. From agent 1's perspective, agent 1's bundle has utility $((q+3)K + 2K) + (q+2)(4K) = (5q+13)K$, agent 2's bundle has utility $((q+1)K + 2K) + (q+4)(4K) = (5q+19)K$, and a most valuable good in agent 2's bundle (e.g., g_{2q+6}) has utility $6K$, so agent 1 is EF1 towards agent 2. From agent 2's perspective, agent 2's bundle has utility $(q+1)K + 2K = (q+3)K$, agent 1's bundle has utility $(q+3)K + 2K = (q+5)K$, and a most valuable good in agent 1's bundle (e.g., g_{2q+5}) has utility $2K$, so agent 2 is EF1 towards agent 1. Accordingly, (A_1, A_2) is an EF1 allocation with size vector $(q+2, q+4)$.

($\Rightarrow$) Let (A_1, A_2) be an EF1 allocation with size vector $\vec{s}$. From agent 1's perspective, $u_1(M) = (10q+32)K$ and a most valuable good (e.g., g_{2q+5}) has utility $6K$. For agent 1 to be EF1 towards agent 2, we must have $u_1(A_1) \geq ((10q+32)K - 6K)/2 = (5q+13)K$ and $u_2(A_1) = u_1(A_1) - (q+2)(4K) \geq (q+5)K$. This means that $u_1(A_2) = u_1(M) - u_1(A_1) \leq (5q+19)K$ and $u_2(A_2) = u_1(A_2) - (q+4)(4K) \leq (q+3)K$. On the other hand, from agent 2's perspective, $u_2(M) = (2q+8)K$ and a most valuable good has utility $2K$. For agent 2 to be EF1 towards agent 1, we must have $u_2(A_2) \geq ((2q+8)K - 2K)/2 = (q+3)K$ and $u_1(A_2) = u_2(A_2) + (q+4)(4K) \geq (5q+19)K$. This means that $u_2(A_1) = u_2(M) - u_2(A_2) \leq (q+5)K$ and $u_1(A_1) = u_2(A_1) + (q+2)(4K) \leq (5q+13)K$. By combining these inequalities, we conclude that these inequalities are tight, i.e., agent 1's utilities for both agents' bundles are *exactly* $(5q+13)K$ and $(5q+19)K$ respectively so that the sum is $(10q+32)K$, and agent 2's utilities for both agents' bundles are *exactly* $(q+5)K$ and $(q+3)K$ respectively so that the sum is $(2q+8)K$. Additionally, both agents must each have a most valuable good worth $6K$ and $2K$ to them respectively. Without loss of generality, we may assume that $g_{2q+5} \in A_1$ and $g_{2q+6} \in A_2$ (note that g_{2q+1} is also a most valuable good, but we use g_{2q+5} and g_{2q+6} for simplicity).

Since $g_{2q+6} \in A_2$, there are $q+3$ goods in $A_2 \setminus \{g_{2q+6}\}$ and $u_2(A_2 \setminus \{g_{2q+6}\}) = (q+3)K - 2K = (q+1)K$. These goods are chosen from $M_1 = \{g_1, \dots, g_{2q+1}\}$ and $M_0 = \{g_{2q+2}, g_{2q+3}, g_{2q+4}\}$. Recall from the construction that $u_2(g) > K$ for all $g \in M_1$, and $u_2(g) = 0$ for all $g \in M_0$. If $A_2 \setminus \{g_{2q+6}\}$ contains at least $q+1$ goods from M_1 , then $u_2(A_2 \setminus \{g_{2q+6}\}) > (q+1)K$, a contradiction. Therefore, $A_2 \setminus \{g_{2q+6}\}$ contains at most q goods from M_1 , and at least 3 goods from M_0 . Since $|M_0| = 3$, we must have $M_0 \subseteq A_2$. Note that $u_2(M_0) = 0$, so $u_2((A_2 \setminus \{g_{2q+6}\}) \setminus M_0) = (q+1)K$. Therefore, the q goods from M_1 in agent 2's bundle have a total utility of $(q+1)K$. These goods, together with g_{2q+2} , induce the set Y_2 with cardinality $q+1$ and sum $(q+1)K$. Then, $Y_1 = Y \setminus Y_2$ and Y_2 give a required partition of Y . $\blacktriangleleft$

The proof of Theorem 6 suggests that the problem is NP-hard even when the sizes of the two agents' bundles differ by *exactly two*. Note that this problem is in P when the sizes of the agents' bundles differ by *at most one* (in fact, every such instance is a Yes-instance by Proposition 2). For two agents with binary utilities, we shall show later that the decision problem is in P (see Theorem 10).

3.2 Constant Number of Agents

Next, we discuss the complexity of the decision problem for a *constant number of agents*. In this case, we devise a pseudopolynomial-time algorithm for deciding the existence of an EF1 allocation with a given size vector. This algorithm uses dynamic programming to check for such an allocation.

► **Lemma 7.** *Let an instance with n agents and a size vector be given, where n is a constant. Suppose that the utility of each good is an integer, and let $R = \max_{i \in N} u_i(M)$. Then, there exists an algorithm running in time polynomial in m and R that decides whether the instance admits an EF1 allocation with the size vector.*

Proof. The algorithm uses dynamic programming. We construct a table with m columns and L rows, where L will be specified later. The index of each row is represented by a tuple containing $a_{i,j}$, $b_{i,j}$, and c_i for each $i, j \in N$, i.e., $(a_{1,1}, a_{1,2}, \dots, a_{n,n}, b_{1,1}, b_{1,2}, \dots, b_{n,n}, c_1, \dots, c_n)$. The value of $a_{i,j}$ is the utility of agent j 's bundle from agent i 's perspective, i.e., $a_{i,j} = u_i(A_j)$; the value of $b_{i,j}$ is the utility of a most valuable good in agent j 's bundle from agent i 's perspective, i.e., $b_{i,j} = \max_{g \in A_j} u_i(g)$ (note that this value is zero if $A_j = \emptyset$); and the value of c_i is the number of goods in agent i 's bundle. Note that $a_{i,j}, b_{i,j} \in \{0, \dots, R\}$ and $c_i \in \{0, \dots, m\}$, so there are $L = (R+1)^{2n^2} (m+1)^n$ rows, which is polynomial in m and R . An entry in column q represents whether an allocation involving $\{g_1, \dots, g_q\}$ is possible for the tuple representing the row, and is either positive or negative.

Initialize every entry to negative. Consider the n possibilities of adding g_1 to each of the agents' bundles respectively, and set the corresponding entries in the first column of the table to positive. In particular, for each $j \in N$, the row represented by the tuple such that $a_{i,j} = b_{i,j} = u_i(g_1)$ and $c_j = 1$ for all $i \in N$, and zero for all other values in the tuple, has the entry (in the first column) set to positive.

Now, for each $q \in \{2, \dots, m\}$ in ascending order, for each positive entry in column $q-1$, consider the n possibilities of adding g_q into each of the n agents' bundles respectively, and set the corresponding entry for each of these possibilities in column q to positive. Once this procedure is done, consider all positive entries in column m . If some positive entry corresponds to an EF1 allocation with the required size vector, then the instance admits such an EF1 allocation; otherwise, no such allocation exists.

Since n is a constant, the number of entries in the table is polynomial in m and R . At each column, there is a polynomial number of rows with positive entries, and hence the update is polynomial. Finally, checking for a feasible EF1 allocation at the last column can also be done in polynomial time. ◀

We now move to *polynomial-time* algorithms that determine the existence of an EF1 allocation with a given size vector. Recall that such an algorithm exists for two agents with identical utilities (Theorem 5). However, it turns out that such an algorithm does not exist for three or more agents with identical utilities, unless $P = NP$. In particular, we establish the NP-hardness of the decision problem via a reduction from BALANCED MULTI-PARTITION with $p = 2$, an NP-hard problem by Proposition 3.

► **Theorem 8.** *REFORMABILITY is weakly NP-complete for $n \geq 3$ agents with identical utilities, where n is a constant.*

Proof. Clearly, this problem is in NP. The “weak” aspect is demonstrated in Lemma 7. Therefore, it suffices to show that this problem is NP-hard.

To show NP-hardness, we shall reduce from the NP-hard problem BALANCED MULTI-PARTITION with $p = 2$ (see Proposition 3). Let a BALANCED MULTI-PARTITION instance with $p = 2$ be given. Define a fair division instance as follows. There are $n \geq 3$ agents with

identical utilities, and a set of goods $M = \{g_1, \dots, g_{2q}, h_1, \dots, h_n\}$ such that $u(g_j) = x_j$ for $j \in \{1, \dots, 2q\}$ and $u(h_k) = (q+1)K$ for $k \in \{1, \dots, n\}$. The size vector $\vec{s}$ is such that $s_1 = s_2 = q+1$ and $s_k = 1$ for all $k \in \{3, \dots, n\}$. This reduction can be done in polynomial time. We claim that there exists an EF1 allocation with size vector $\vec{s}$ in this instance if and only if X can be partitioned into multisets X_1, X_2 of equal cardinalities and sums.

($\Leftarrow$) Let (X_1, X_2) be such a partition. Define an allocation such that agent k receives h_k for $k \in N$, agent 1 additionally receives the q goods corresponding to the integers in X_1 , and agent 2 additionally receives the q goods corresponding to the integers in X_2 . We show that this allocation is EF1. The utilities of agent 1's and agent 2's bundles are $2(q+1)K$ each, and the utilities of the other agents' bundles are $(q+1)K$ each, so agents 1 and 2 do not envy anyone else. Therefore, it remains to check that agent k is EF1 towards agents 1 and 2 for $k \in \{3, \dots, n\}$. Upon the removal of the single good h_1 (resp. h_2) from agent 1's (resp. agent 2's) bundle, the remaining bundle has utility $(q+1)K$, so agent k is EF1 towards agent 1 (resp. agent 2). Therefore, the allocation is EF1, as desired.

($\Rightarrow$) Let $(A_1, \dots, A_n)$ be an EF1 allocation with size vector $(q+1, q+1, 1, \dots, 1)$. If agent 1's bundle has at least two goods from $\{h_1, \dots, h_n\}$, then her bundle without the most valuable good has utility more than $(q+1)K$ since her bundle also contains other goods with positive utility. Agent 3, having a bundle of utility at most $(q+1)K$, will not be EF1 towards agent 1, contradicting the assumption that the allocation is EF1. Therefore, agent 1's bundle has at most one good from $\{h_1, \dots, h_n\}$; likewise for agent 2's bundle. This means that every agent receives *exactly* one good from $\{h_1, \dots, h_n\}$. Having established this, agent 3's bundle has a utility of $(q+1)K$, and agent 3 is EF1 towards agent 1. This means that agent 1's bundle without a most valuable good (say, some h_k) must have utility at most $(q+1)K$. The same argument can be used to show the same statement for agent 2's bundle. This means that the goods $\{g_1, \dots, g_{2q}\}$ must be divided between agents 1 and 2 with each agent receiving a utility of *exactly* $(q+1)K$. Such a division of $\{g_1, \dots, g_{2q}\}$ induces a partition of X into two multisets of equal cardinalities and sums, as desired. $\blacktriangleleft$

Since the decision problem is NP-hard even for identical utilities, it must also be NP-hard for general utilities. We now consider another class of utilities: binary utilities. When there are n agents, every good g belongs to one of 2^n types of goods represented by the vector $(u_1(g), \dots, u_n(g))$. For the purpose of determining whether an EF1 allocation exists, it suffices to consider different goods of the same type as *indistinguishable*. We say that two allocations are in the same equivalence class if the number of goods of each type that each agent has is the same in both allocations. If $\mathcal{A}$ is an EF1 allocation, then all allocations in the same equivalence class as $\mathcal{A}$ are also EF1 and are reachable from $\mathcal{A}$. We shall proceed with a result which enumerates all (essentially equivalent) EF1 allocations in time polynomial in the number of goods, provided that the number of agents is a constant.

► **Lemma 9.** *Let an instance with n agents with binary utilities and a size vector be given, where n is a constant. Then, there exists an algorithm running in time polynomial in m that outputs all equivalence classes of EF1 allocations with the size vector.*

Proof. An agent's bundle can be represented by a 2^n -vector where each component of the vector is the number of goods of that type in her bundle. Since the number of goods of each type is an integer between 0 and m , there are $m+1$ possible values for each component, and hence at most $(m+1)^{2^n}$ possible vectors to represent each agent's bundle. Allocations in an equivalence class can be represented by an ordered collection of n such vectors – one for each agent – and there are at most $((m+1)^{2^n})^n$ such collections. Since $((m+1)^{2^n})^n$ is polynomial in m whenever n is a constant, there is at most a polynomial number of possible

equivalence classes of allocations in the instance. For each of these equivalence classes of allocations, we can check whether an allocation in the equivalence class is EF1 and has the required size vector in polynomial time, and output the equivalence class if so. Therefore, the overall running time is polynomial in m , as claimed. $\blacktriangleleft$

Lemma 9 implies that the decision problem can be solved efficiently for binary utilities.

► **Theorem 10.** *REFORMABILITY is in P for a constant number of agents with binary utilities.*

3.3 General Number of Agents

For any constant number of agents, the problem of determining the existence of an EF1 allocation with a given size vector is *weakly* NP-hard by Theorem 8 (even for identical utilities). For a general number of agents, the pseudopolynomial-time algorithm as described in Lemma 7 does not work, since that algorithm is at least exponential in the number of agents. Therefore, the decision problem for a general number of agents might not be *weakly* NP-hard. We show that the problem is indeed *strongly* NP-hard, even for identical utilities, by a reduction from 3-PARTITION, a strongly NP-hard problem [16, p. 224].

► **Theorem 11.** *REFORMABILITY is strongly NP-complete for identical utilities.*

The reduction in Theorem 11 requires us to check whether the set of agents holding the goods with utilities equal to the integers in the partition problem is EF1 towards the remaining agents. This stands in contrast to the reduction in Theorem 8, which entails checking whether the complementary set of agents is EF1 towards the remaining agents.

For binary utilities, the decision problem for a *constant* number of agents is in P (Theorem 10). The crucial reason is that in this case, the number of different types of goods is also a constant, which allows us to enumerate all the (essentially equivalent) EF1 allocations in polynomial time (Lemma 9). This is no longer possible when the number of agents is non-constant. In fact, we show that the decision problem is NP-hard for a general number of agents with binary utilities. To this end, we reduce from GRAPH k -COLORABILITY, which is NP-hard for any fixed $k \geq 3$ [16, p. 191].

► **Theorem 12.** *REFORMABILITY is NP-complete for binary utilities.*

Proof. Clearly, this problem is in NP. Therefore, it suffices to show that it is NP-hard.

To this end, we shall reduce from GRAPH k -COLORABILITY with $k \geq 3$. In GRAPH k -COLORABILITY, we are given a graph $G = (V, E)$ and a positive integer k , and the problem is to decide whether G is k -colorable, i.e., whether each of the vertices in V can be assigned one of k colors in such a way that no two adjacent vertices are assigned the same color. This decision problem is known to be NP-hard for any fixed $k \geq 3$ [16, p. 191].

Let an instance of GRAPH k -COLORABILITY be given with a fixed $k \geq 3$, where $V = \{v_1, \dots, v_p\}$ and $E = \{e_1, \dots, e_q\}$. Define a fair division instance as follows. There are $n = q + k$ agents where the first q agents are called *edge agents* and the last k agents are called *color agents*. There are $m = kp$ goods. Each color agent assigns zero utility to every good. For $r \in \{1, \dots, q\}$, if $e_r = \{v_i, v_j\}$, then the r^{th} edge agent assigns a utility of 1 to each of g_i and g_j , and zero utility to every other good. Note that only the first p goods correspond to vertices and are valuable to the edge agents whose corresponding edges are incident to the vertices; the remaining $(k-1)p$ goods are not valuable to any agent. The size vector $\vec{s}$ is such that $s_r = 0$ for each edge agent r and $s_c = p$ for each color agent c . This reduction can be done in polynomial time. We claim that there exists an EF1 allocation with size vector $\vec{s}$ in this instance if and only if G is k -colorable.

($\Leftarrow$) Let a proper k -coloring of G be given. For $t \in \{1, \dots, p\}$, if vertex v_t is assigned the color c , then allocate g_t to the color agent c . Since there are p such goods and each color agent is supposed to have p goods in her bundle, it is possible to allocate all of these goods. Subsequently, allocate the remaining goods arbitrarily among the color agents until every color agent has exactly p goods. We claim that this allocation is EF1. Every color agent assigns zero utility to every good and is thus EF1 towards every other agent. Each edge agent assigns a utility of 1 to only two goods, so we only need to check that these two goods are in different bundles. Indeed, these two goods correspond to vertices which are adjacent to each other in G , and proper coloring of G implies that the vertices are of different colors, so the corresponding goods are in different color agents' bundles. Hence, the allocation is EF1.

($\Rightarrow$) Let an EF1 allocation with size vector $\vec{s}$ be given. For $t \in \{1, \dots, p\}$, if the good g_t is with color agent c , assign v_t to color c . We claim that this coloring is a proper k -coloring of G . Since there are k color agents, at most k colors are used. Therefore, it suffices to check that adjacent vertices are assigned different colors. Let $v_i, v_j \in V$ be adjacent vertices. Then, there exists an edge $e_r = \{v_i, v_j\}$. The edge agent r assigns a utility of 1 to each of g_i and g_j . Since agent r 's bundle is empty, g_i and g_j must be in different (color agents') bundles in order for agent r to be EF1 towards every other agent. This implies that v_i and v_j are assigned different colors. $\blacktriangleleft$

Even though the decision problem is NP-hard for identical *or* binary utilities, we prove next that it can be solved in polynomial time for identical *and* binary utilities. Indeed, this can be done by checking whether the total number of valuable goods is within a certain threshold which can be computed in polynomial time. This threshold is in fact the sum over all agents $j \in N$ of the minimum between s_j and $1 + \min_{i \in N} s_i$. If the number of valuable goods is within this threshold, then we can distribute the valuable goods in a round-robin fashion first, thereby ensuring EF1. Conversely, if the number of valuable goods is beyond this threshold, then some agent i with the minimum s_i will not be EF1 towards another agent who receives at least $s_i + 2$ valuable goods.

► **Theorem 13.** *REFORMABILITY is in P for identical binary utilities.*

4 Optimal Number of Exchanges

In this section, we consider the complexity of computing the *optimal number of exchanges* required to reach an EF1 allocation from an initial allocation.

Recall that the decision problem in Section 3 is to determine whether there exists an EF1 allocation that can be reached from a given initial allocation. This is equivalent to determining whether the optimal number of exchanges to reach an EF1 allocation is finite or not. We have established a few scenarios where there exist polynomial-time algorithms for this task: (a) two agents with identical utilities (Theorem 5), (b) any constant number of agents with binary utilities (Theorem 10), and (c) any number of agents with identical binary utilities (Theorem 13). For these scenarios, we can run the respective polynomial-time algorithms to decide whether such an EF1 allocation exists – if none exists, then the optimal number of exchanges is ∞ . Therefore, for the proofs in this section pertaining to these scenarios, we proceed with the assumption that such an EF1 allocation exists. We will show that the problem of computing the optimal number of exchanges for these scenarios is also in P; our algorithms can be modified to compute an optimal *sequence* of exchanges as well.

For the remaining scenarios where the decision problem in Section 3 is NP-hard, it is NP-hard to even decide whether the optimal number of exchanges to reach an EF1 allocation is finite or not. Therefore, for these scenarios, we shall focus on the special case where the

given size vector is *balanced*, so that the optimal number of exchanges is guaranteed to be finite (see Proposition 2). Even with this assumption, we will show that the computational problem for these scenarios remains NP-hard.

For convenience, we refer to as **OPTIMAL EXCHANGES** the problem of deciding – given an instance, an initial allocation in the instance, and a number k – whether the optimal number of exchanges required to reach an EF1 allocation is at most k . Note that the optimal number of exchanges in the scenarios mentioned above are finite. By the proof of Proposition 1, the optimal number of exchanges in these scenarios is polynomial in the number of agents and the number of goods. As a result, **OPTIMAL EXCHANGES** is in NP regardless of the number of agents, as we can easily verify whether an exchange path starting from the given initial allocation indeed reaches an EF1 allocation using at most k exchanges.

We begin with the case of two agents. For two agents with identical utilities, we show that there exists a polynomial-time algorithm that computes the optimal number of exchanges. This algorithm performs the exchanges until an EF1 allocation is reached, while keeping track of the number of exchanges required. The algorithm is “greedy” in the sense that at each step, it performs an exchange involving the most valuable good from the agent whose bundle has the higher utility, and the least valuable good from the other agent. We demonstrate that this choice is the best for the number of exchanges required to reach an EF1 allocation.

► **Theorem 14.** *OPTIMAL EXCHANGES is in P for two agents with identical utilities.*

Proof. We show that we can compute the optimal number of exchanges in polynomial time. We assume that an EF1 allocation with the given size vector exists. If the initial allocation $\mathcal{A}$ is EF1, we are done. Otherwise, assume without loss of generality that agent 2 has a higher utility than agent 1 in $\mathcal{A}$. Let $\vec{s} = (s_1, s_2)$ be the size vector. By rearranging the labels of the goods, assume that the goods are in non-increasing order of utility, i.e., $u(g_1) \geq u(g_2) \geq \dots \geq u(g_m)$. The algorithm proceeds as follows: repeatedly exchange the most valuable good in agent 2’s bundle with the least valuable good in agent 1’s bundle until agent 1 is EF1 towards agent 2. The optimal number of exchanges required is then the number of exchanges made in this algorithm.

First, we claim that in each exchange, a good from $\{g_1, \dots, g_{s_1}\}$ in agent 2’s bundle is always exchanged with a good from $\{g_{s_1+1}, \dots, g_m\}$ in agent 1’s bundle. Suppose on the contrary that this is not true. Let $\mathcal{A}'$ be the allocation just before we make the exchange that violates this claim. The only way for the claim to be violated is when $A'_1 = \{g_1, \dots, g_{s_1}\}$ and $A'_2 = \{g_{s_1+1}, \dots, g_m\}$. If $s_1 \leq s_2$, then by Lemma 4, there does not exist an EF1 allocation with size vector $\vec{s}$ – this would contradict our assumption that an EF1 allocation with size vector $\vec{s}$ exists. Otherwise, $s_1 > s_2$, and every good in A'_1 has a higher utility than every good in A'_2 , so agent 1 is EF1 towards agent 2. This would contradict our assumption that the algorithm has not terminated. Hence, the claim is true.

By the claim in the previous paragraph, the total number of exchanges made is at most $\min\{s_1, s_2\} \leq m$. Each exchange can be performed in polynomial time, and so the algorithm terminates in polynomial time. We show next that an EF1 allocation is obtained when the algorithm terminates. It suffices to show that agent 2 is EF1 towards agent 1 in the final allocation. To this end, we show that agent 2 is EF1 towards agent 1 at every step of the algorithm. Let the initial allocation be $\mathcal{A}^0 = \mathcal{A}$, and let $\mathcal{A}^t$ be the allocation after t steps of the algorithm. Note that $\mathcal{A}^0$ satisfies the condition that agent 2 is EF1 towards agent 1, since agent 2 has a higher utility than agent 1 in $\mathcal{A}$. We show that if $\mathcal{A}^t$ has the property that agent 2 is EF1 towards agent 1 and agent 1 is *not* EF1 towards agent 2, then $\mathcal{A}^{t+1}$ has the

property that agent 2 is EF1 towards agent 1. Suppose that $g \in A_2^t$ is exchanged with $h \in A_1^t$; note that g is a good with the highest utility in A_2^t . This means that $u(A_1^t) < u(A_2^t \setminus \{g\})$. Then,

$$u(A_2^{t+1}) \geq u(A_2^{t+1} \setminus \{h\}) = u(A_2^t \setminus \{g\}) > u(A_1^t) \geq u(A_1^t \setminus \{h\}) = u(A_1^{t+1} \setminus \{g\}),$$

showing that agent 2 is EF1 towards agent 1 in $\mathcal{A}^{t+1}$.

Finally, we show that the optimal number of exchanges required to reach an EF1 allocation is at least the number of exchanges made in this algorithm. Let T be the number of exchanges made in this algorithm. For each $t \in \{1, \dots, T\}$, let $g^t \in A_2$ (resp. $h^t \in A_1$) be the good in agent 2's bundle (resp. agent 1's bundle) that is exchanged at the t^{th} step of the algorithm. Note that $u(g^1) \geq \dots \geq u(g^T) \geq u(h^T) \geq \dots \geq u(h^1)$. Also, we have $u(A_1^{T-1}) < u(A_2^{T-1} \setminus \{g^T\})$, where g^T is a good with the highest utility in agent 2's bundle in $\mathcal{A}^{T-1}$. Suppose on the contrary that only $k \leq T - 1$ exchanges are required to reach an EF1 allocation. Since $\mathcal{A}$ is not EF1, we have $1 \leq k < T$. Let (B_1, B_2) be the EF1 allocation after the k exchanges. The utility of B_1 is upper-bounded by the utility of A_1 after adding k goods of the highest utility from A_2 and removing k goods of the lowest utility from A_1 , so

$$\begin{aligned} u(B_1) &\leq u((A_1 \cup \{g^1, \dots, g^k\}) \setminus \{h^1, \dots, h^k\}) \\ &= u(A_1) + \sum_{t=1}^k (u(g^t) - u(h^t)) \\ &\leq u(A_1) + \sum_{t=1}^{T-1} (u(g^t) - u(h^t)) \\ &= u((A_1 \cup \{g^1, \dots, g^{T-1}\}) \setminus \{h^1, \dots, h^{T-1}\}) = u(A_1^{T-1}). \end{aligned}$$

On the other hand, the utility of B_2 without the most valuable good is lower-bounded by the utility of A_2 after adding k goods of the lowest utility from A_1 and removing $k + 1 \leq T$ goods of the highest utility from A_2 , so we have

$$\begin{aligned} u(B_2 \setminus \{g\}) &\geq u((A_2 \cup \{h^1, \dots, h^k\}) \setminus \{g^1, \dots, g^{k+1}\}) \\ &= u(A_2) - u(g^1) - \sum_{t=1}^k (u(g^{t+1}) - u(h^t)) \\ &\geq u(A_2) - u(g^1) - \sum_{t=1}^{T-1} (u(g^{t+1}) - u(h^t)) \\ &= u((A_2 \cup \{h^1, \dots, h^{T-1}\}) \setminus \{g^1, \dots, g^T\}) = u(A_2^{T-1} \setminus \{g^T\}) \end{aligned}$$

for every $g \in B_2$. This gives the inequality $u(B_1) \leq u(A_1^{T-1}) < u(A_2^{T-1} \setminus \{g^T\}) \leq u(B_2 \setminus \{g\})$ for all $g \in B_2$. Hence, agent 1 is not EF1 towards agent 2 in (B_1, B_2) , contradicting the assumption that (B_1, B_2) is EF1. It follows that at least T exchanges are required to reach an EF1 allocation. $\blacktriangleleft$

However, if the utilities are not identical, then computing the optimal number of exchanges is NP-hard, even for balanced allocations. To show this hardness, we modify the construction from the proof of Theorem 6.

► **Theorem 15.** *OPTIMAL EXCHANGES is NP-complete for two agents, even when the initial allocation is balanced.*

Proof. Clearly, this problem is in NP. To demonstrate NP-hardness, we modify the construction from the proof of Theorem 6. Recall that we have $Y = \{y_1, \dots, y_{2q+2}\}$ with $K < y_j \leq 2K$ for $j \in \{1, \dots, 2q\}$, $y_{2q+1} = 2K$, and $y_{2q+2} = 0$. A fair division instance $\mathcal{I}'$ is defined with $n = 2$ agents and a set of goods $M' = \{g_1, \dots, g_{2q+6}\}$. Agent 2's utility is such that $u_2(g_j) = y_j$ for $j \in \{1, \dots, 2q+2\}$, $u_2(g_{2q+3}) = u_2(g_{2q+4}) = 0$, and $u_2(g_{2q+5}) = u_2(g_{2q+6}) = 2K$. Agent 1's utility is such that $u_1(g) = u_2(g) + 4K$ for all $g \in M$. The size vector $\vec{s}'$ is $(q+2, q+4)$. In Theorem 6, it was proven that there exists an EF1 allocation with size vector $\vec{s}'$ in this instance if and only if Y can be partitioned into two multisets Y_1 and Y_2 of equal cardinalities (i.e., of size $q+1$ each) with sums $(q+3)K$ and $(q+1)K$ respectively. Both problems were proven to be NP-hard.

Define a new fair division instance $\mathcal{I}$ as follows. There are $n = 2$ agents and a set of goods $M = \{g_1, \dots, g_{4q+12}\}$. For $j \in \{1, \dots, 2q+6\}$, the utility of g_j for each agent is identical to that in the original fair division instance $\mathcal{I}'$. For $j \in \{2q+7, \dots, 4q+12\}$, we have $u_i(g_j) = 0$ for $i \in \{1, 2\}$. The size vector $\vec{s}$ is $(2q+6, 2q+6)$. In the initial allocation $\mathcal{A}$, agent 1 has $A_1 = \{g_{2q+7}, \dots, g_{4q+12}\}$ and agent 2 has $A_2 = \{g_1, \dots, g_{2q+6}\}$. This reduction can be done in polynomial time. We claim that the optimal number of exchanges required to reach an EF1 allocation from $\mathcal{A}$ is at most $q+2$ in $\mathcal{I}$ if and only if there exists an EF1 allocation with size vector $\vec{s}'$ in $\mathcal{I}'$.

($\Leftarrow$) Let (A'_1, A'_2) be an EF1 allocation with size vector $\vec{s}'$ in $\mathcal{I}'$. Note that $A'_1 \subseteq A_2$ and $|A'_1| = q+2$. In $\mathcal{A}$, exchange the $q+2$ goods from A'_1 with any $q+2$ goods in A_1 . This requires a total of $q+2$ exchanges. The new allocation has exactly the same goods as that in (A'_1, A'_2) along with other goods with zero utility, and so is EF1. Therefore, the optimal number of exchanges to reach an EF1 allocation from $\mathcal{A}$ is at most $q+2$.

($\Rightarrow$) Suppose that an EF1 allocation $\mathcal{B}$ is reached from $\mathcal{A}$ after $t \leq q+2$ exchanges in $\mathcal{I}$. We may assume that every good is not exchanged more than once. By the same reasoning as in the proof of Theorem 6, we must have $u_1(B_1) \geq (5q+13)K$ and $u_2(B_1) \leq (q+5)K$. Since t goods are transferred from A_2 , we have $u_1(B_1) = u_2(B_1) + t(4K) \leq (q+5)K + (q+2)(4K) = (5q+13)K$. This means that the inequalities for $u_1(B_1)$ are tight, and we have $u_1(B_1) = 5q+13$ and $t = q+2$. Letting $M_1 = A_2 \cap B_1$, we have $|M_1| = q+2$ and $M_1 \subseteq A_2 \subseteq M'$. Since (B_1, B_2) is an EF1 allocation, the allocation that removes all goods with zero utility is also EF1, namely, $(M_1, M' \setminus M_1)$. This induces an EF1 allocation with size vector $\vec{s}'$ in $\mathcal{I}'$. $\blacktriangleleft$

While a polynomial-time algorithm to compute the optimal number of exchanges exists for two agents with identical utilities, this is not the case for three or more agents unless $P = NP$. Indeed, we establish the NP-hardness of this problem via a reduction from BALANCED MULTI-PARTITION with $p \geq 2$, an NP-hard problem by Proposition 3.

► **Theorem 16.** *OPTIMAL EXCHANGES is NP-complete for $n \geq 3$ agents with identical utilities, where n is a constant, even when the initial allocation is balanced.*

Next, we consider binary utilities. We have shown that deciding whether the optimal number of exchanges to reach an EF1 allocation is finite can be done in polynomial time (Theorem 10). We now establish that the same is true for computing this exact number.

► **Theorem 17.** *OPTIMAL EXCHANGES is in P for a constant number of agents with binary utilities.*

Let us now consider a non-constant number of agents. We have shown that computing the optimal number of exchanges required to reach an EF1 allocation is NP-hard, even for identical utilities (Theorem 16). We thus consider binary utilities.

Although this problem belongs to P when the number of agents is a constant (Theorem 17), we show that it is NP-hard for a general number of agents, even for the special case where the initial allocation is balanced. To this end, we reduce from EXACT COVER BY 3-SETS.

► **Theorem 18.** *OPTIMAL EXCHANGES is NP-complete for binary utilities, even when the initial allocation is balanced.*

Finally, we consider identical binary utilities. We show that for this class of utilities, the computational problem can be solved efficiently regardless of whether the size vector of the initial allocation is balanced or not. To show this, we demonstrate that the greedy algorithm allows an EF1 allocation to be reached using the smallest number of exchanges.

► **Theorem 19.** *OPTIMAL EXCHANGES is in P for identical binary utilities.*

5 Worst-Case Bounds

In this section, instead of instance-specific optimization, we turn our attention to the *worst-case* number of exchanges required to reach an EF1 allocation from some initial allocation. Since an EF1 allocation may not always be reachable (as can be seen from Section 3), we shall focus on the special case where the number of goods in each agent's bundle is the same. We say that a size vector $\vec{s} = (s_1, \dots, s_n)$ is *s-balanced* for a positive integer s if $s_i = s$ for all $i \in N$, and an allocation is *s-balanced* if it has an *s-balanced* size vector. We shall consider the worst-case number of exchanges starting from an *s-balanced* allocation for n agents.

Given n and s , let $f(n, s)$ be the smallest integer such that for every instance with n agents and ns goods and every *s-balanced* allocation $\mathcal{A}$ in the instance, there exists an EF1 allocation that can be reached from $\mathcal{A}$ using at most $f(n, s)$ exchanges. We shall examine the bounds for $f(n, s)$.

We first derive an upper bound for $f(n, s)$. At a high level, we use an algorithm by Biswas and Barman [4] to find an EF1 allocation under cardinality constraints such that every agent retains roughly s/n of her goods from her original bundle. The algorithm also distributes the goods in each agent's initial bundle to the other agents as evenly as possible in order to maximize the number of goods that can be exchanged one-to-one, thereby minimizing the total number of exchanges. One can check that roughly $s(n-1)/2$ exchanges are required to reach this EF1 allocation from the initial allocation.

► **Theorem 20.** *Let n and s be positive integers, and let $q = \lfloor s/n \rfloor$ and $r = s - qn$ be the quotient and remainder when s is divided by n respectively. Then,*

$$f(n, s) \leq \begin{cases} s(n-1)/2 & \text{if } r = 0; \\ s(n-1)/2 + r(n-3)/2 + 1 & \text{otherwise.} \end{cases}$$

Moreover, we have $f(2, s) \leq (s-r)/2$ for all s .

Proof. Let $\mathcal{A}$ be an *s-balanced* allocation. It suffices to find an EF1 *s-balanced* allocation $\mathcal{B}$ such that the optimal number of exchanges to reach $\mathcal{B}$ from $\mathcal{A}$ is at most the expression given in the theorem statement.

When $n = 2$, allocate the goods in A_1 to the two agents in a round-robin fashion with agent 1 going first, and allocate the goods in A_2 to the two agents in a round-robin fashion with agent 2 going first. Call this new allocation $\mathcal{B}$. Note that $\mathcal{B}$ is clearly *s-balanced*. We have $A_i \cap B_{3-i} = (s-r)/2$ for $i \in \{1, 2\}$, so the optimal number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}$ is (exactly) $(s-r)/2$. To see that $\mathcal{B}$ is EF1, observe that agent 1 does not envy agent 2 with respect to the goods chosen from A_1 and is EF1 towards agent 2 with respect to the goods chosen from A_2 , so agent 1 is EF1 towards agent 2 in $\mathcal{B}$; likewise, agent 2 is EF1 towards agent 1 in $\mathcal{B}$. This shows that $f(2, s) \leq (s-r)/2$.

When $n \geq 3$, we shall find an EF1 s -balanced allocation $\mathcal{B}$ by generalizing the method for two agents. We define $n + r$ categories of goods $C_1, \dots, C_n, D_1, \dots, D_r$ as follows. For $i \in N$, category C_i contains qn goods arbitrarily selected from A_i only; note that r goods remain unselected in A_i . Next, we form D_w recursively as follows: let $w \in \{1, \dots, r\}$ be the smallest index such that D_w does not have n goods yet, let $i \in N$ be the smallest index such that A_i still has unselected goods, arbitrarily select a good in A_i , and add it to D_w . At the end of this process, every category C_i has exactly qn goods from A_i , and every category D_w has exactly n goods from consecutive agents' bundles, say, $A_{i_w}, A_{i_w+1}, \dots, A_{j_w}$.

We now proceed to form $\mathcal{B}$ using the algorithm by Biswas and Barman [4] which finds an EF1 allocation under cardinality constraints. In particular, there exists an EF1 allocation $\mathcal{B} = (B_1, \dots, B_n)$ such that $|C_i \cap B_j| = |C_i|/n = q$ for all $i, j \in N$ and $|D_w \cap B_j| = |D_w|/n = 1$ for all $w \in \{1, \dots, r\}, j \in N$. Also, $\mathcal{B}$ is s -balanced because $|B_j| = qn + r = s$ for all $j \in N$. We shall bound the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}$.

For each unordered pair of distinct $i, j \in N$, exchange the q goods from $C_i \cap B_j$ (which are in A_i) with the q goods from $C_j \cap B_i$ (which are in A_j). This requires a total of $qn(n-1)/2$ exchanges. Call this intermediate allocation $\mathcal{A}' = (A'_1, \dots, A'_n)$. At this point, the only goods that are possibly in the wrong bundles in $\mathcal{A}'$ (as compared to $\mathcal{B}$) are the goods in all the D_w , and there are at most rn such goods. For each $i \in N$, let $X_i = A'_i \cap (D_1 \cup \dots \cup D_r)$.

If $r = 0$, then $\mathcal{A}' = \mathcal{B}$, and we are done since the total number of exchanges is $qn(n-1)/2 = s(n-1)/2$. Else, $r > 0$. Consider the directed graph where the vertices are the agents and each edge e_g represents a good $g \in M$ such that if $g \in A'_i \cap B_j$, then $e_g = (i, j)$. Igarashi et al. [22, Prop. 4.1] showed that the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}'$ is $m - c^*$, where c^* is the maximum possible cardinality of a partition of the edges of the graph into (directed) circuits. In $\mathcal{A}'$, qn^2 goods from all the C_i are in the correct bundle by the previous process, and each of the edges representing these goods has its own circuit, say, (i, i) if the good is in A'_i . We shall show that the edges representing the rn goods in all the D_w can be partitioned into at least $2r - 1$ disjoint circuits. This will give at least $qn^2 + (2r - 1) = sn - (rn - 2r + 1)$ as the cardinality of one such partition of the edges of the graph into circuits. Accordingly, $c^* \geq sn - (rn - 2r + 1)$, and the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}'$ is $m - c^* \leq rn - 2r + 1$. Then, the number of exchanges required to reach $\mathcal{B}$ from $\mathcal{A}$ (via $\mathcal{A}'$) is at most $qn(n-1)/2 + (rn - 2r + 1) = s(n-1)/2 + r(n-3)/2 + 1$, establishing the theorem.

Let $w \in \{1, \dots, r\}$ be given. We shall show that there exists a cycle formed with a subset of the edges representing the goods in D_w . The goods in D_w come from consecutive agents' bundles in $\mathcal{A}'$, say, agents i_w to j_w . Every agent receives exactly one good from D_w in $\mathcal{B}$; in particular, agents i_w to j_w receive exactly one good from D_w each. Consider the good g in $D_w \cap B_{i_w}$. If g is in X_{i_w} , then the edge $e_g = (i_w, i_w)$ is a desired cycle. Otherwise, g belongs to some agent $i' \in \{i_w + 1, \dots, j_w\}$ in $\mathcal{A}'$. Then, the edge e_g is (i', i_w) . Next, we consider the good g' in $D_w \cap B_{i'}$, and find the agent that has g' in $\mathcal{A}'$. The edge representing g' then points to i' from that agent. By repeating this, we eventually find a cycle formed with some of these edges and with a subset of the agents i_w to j_w as vertices. Let $M_w \subseteq D_w$ be the set of goods that are represented by the edges in this cycle. Note that each X_i for $i \in \{i_w, \dots, j_w\}$ contains at most one good in M_w , and each X_i for $i \in N \setminus \{i_w, \dots, j_w\}$ does not contain any good in M_w .

Now, consider the goods represented by the edges of the r cycles – one for each w . Note that these cycles are disjoint since the sets M_w are pairwise disjoint. Let $M_0 = \bigcup_{w=1}^r M_w$. We claim that $|M_0| < 2n$. Since the r goods in X_1 are entirely contained in D_1 , we have $|X_1 \cap M_1| \leq 1$ and $|X_1 \cap M_w| = 0$ for $w \in \{2, \dots, r\}$, which implies that $|\bigcup_{w=1}^r (X_1 \cap M_w)| \leq 1$.

Now, for each $i \in N \setminus \{1\}$, the r goods in X_i can only be contained in at most two D_w – to see this, observe that if the r goods are contained in $D_{w'}$, $D_{w'+1}$, and $D_{w'+2}$, then $D_{w'+1} \subseteq X_i$, which implies that $r = |X_i| \geq |D_{w'+1}| = n$, a contradiction. Thus, we have $|X_i \cap M_w| \leq 1$ for all $w \in \{1, \dots, r\}$, and $|X_i \cap M_w| = 1$ for at most two w , and so $|\bigcup_{w=1}^r (X_i \cap M_w)| \leq 2$. Since $M_0 = \bigcup_{i \in N} \bigcup_{w=1}^r (X_i \cap M_w)$, we have $|M_0| \leq 1 + (n-1) \cdot 2 < 2n$, proving the claim.

Finally, consider the edges representing the rn goods in all the D_w . We have shown that fewer than $2n$ of these edges can be used to form r disjoint circuits (in fact, cycles). There are more than $rn - 2n = (r-2)n$ edges remaining. Since we can always require every circuit to have length at most n , there exists a partition of the remaining edges into more than $r-2$ disjoint circuits, i.e., at least $r-1$ disjoint circuits. The total number of circuits in this partition is at least $r + (r-1) = 2r-1$. This completes the proof. ◀

If no good is involved in more than one exchange, then $s(n-1)/2$ exchanges means that a total of $s(n-1) = m(1-1/n)$ goods are exchanged. When n is large, the fraction of goods involved in the exchanges becomes close to 1. While this bound might not seem impressive, we show next that it is, in fact, already essentially tight. Specifically, we establish a lower bound for $f(n, s)$ by constructing an instance (with binary utilities) and an s -balanced allocation $\mathcal{A}$ in the instance such that roughly $s(n-1)/2$ exchanges are necessary to reach an EF1 allocation from $\mathcal{A}$.

► **Theorem 21.** *Let n and s be positive integers, and let $q = \lfloor s/n \rfloor$ and $r = s - qn$ be the quotient and remainder when s is divided by n respectively. Then,*

$$f(n, s) \geq \begin{cases} s(n-1)/2 & \text{if } r = 0; \\ s(n-1)/2 - (n-r)/2 & \text{otherwise.} \end{cases}$$

For two agents, Theorems 20 and 21 give a tight bound of $f(2, s) = (s-r)/2 = m/4 - r/2 = \lfloor m/4 \rfloor$. This means that in the worst-case scenario, the number of exchanges required to reach an EF1 allocation is roughly one-quarter of the total number of goods between the two agents, or equivalently, roughly half of the goods need to be exchanged between the two agents to reach an EF1 allocation.

Theorems 20 and 21 also give a tight bound of $f(n, s) = s(n-1)/2$ whenever s is divisible by n . By observing the proof of Theorem 20, we can achieve an EF1 allocation with $f(n, s)$ exchanges without involving each good in more than one exchange. This means that a $(1-1/n)$ fraction of all goods need to be exchanged in the worst-case scenario. Intuitively, this happens when each agent only values the goods in the bundle of every agent except her own in the initial allocation, and therefore needs to ensure that these goods are evenly distributed among all agents including herself.

Define $f_{\text{bin}}(n, s)$ as the smallest integer such that for every *binary* instance with n agents and ns goods and every s -balanced allocation $\mathcal{A}$ in the instance, there exists an EF1 allocation that can be reached from $\mathcal{A}$ using at most $f_{\text{bin}}(n, s)$ exchanges. The proof of Theorem 21 uses a binary instance, which means that the lower bound of the theorem works for f_{bin} as well. Clearly, the upper bound of Theorem 20 works for f_{bin} , so the discussion in the preceding paragraphs also applies to binary instances too.

Given n and s , let $f_{\text{id,bin}}(n, s)$ be the smallest integer such that for every instance with n agents with *identical binary* utilities and ns goods and every s -balanced allocation $\mathcal{A}$ in the instance, there exists an EF1 allocation that can be reached from $\mathcal{A}$ using at most $f_{\text{id,bin}}(n, s)$ exchanges. We show that $f_{\text{id,bin}}(n, s)$ is roughly $sn/4$ for even n and $s(n-1)(n+1)/4n$ for odd n – note that this is approximately half of the bound $f(n, s)$ for arbitrary utilities, which

is roughly $s(n-1)/2$ as seen earlier. The upper bounds (of $sn/4$ and $s(n-1)(n+1)/4n$ respectively) correspond to the case where half of the agents have all the valuable goods while the remaining half have all the non-valuable goods.

► **Theorem 22.** *Let n and s be positive integers. If n is even, then $\frac{n}{2} \lfloor \frac{s}{2} \rfloor \leq f_{\text{id},\text{bin}}(n, s) \leq \frac{sn}{4}$. If n is odd, then $\frac{n+1}{2} \lfloor \frac{s(n-1)}{2n} \rfloor \leq f_{\text{id},\text{bin}}(n, s) \leq \frac{s(n-1)(n+1)}{4n}$.*

6 Conclusion and Future Work

We have studied the reformability of unfair allocations and the number of exchanges required in the reformation process. We uncovered several distinctions in the complexity of these problems based on the number of agents and their utility functions, and showed that the number of exchanges required to reach an EF1 allocation is relatively high in the worst case.

While our worst-case bounds for general utilities are already exactly tight in certain scenarios and almost tight generally, an open question is to tighten them for more than two agents when the number of goods in each agent's bundle is not divisible by the number of agents. Another interesting direction is to require each exchange to be beneficial for both agents involved. One could also consider the model of *transferring* goods instead of *exchanging* them – an EF1 allocation is always reachable from any allocation in this model, so a natural question is to determine the optimal number of exchanges needed for this goal.

References

- 1 Haris Aziz, Péter Biró, Jérôme Lang, Julien Lesca, and Jérôme Monnot. Efficient reallocation under additive and responsive preferences. *Theoretical Computer Science*, 790:1–15, 2019. doi:10.1016/J.TCS.2019.05.011.
- 2 Haris Aziz, Ildikó Schlotter, and Toby Walsh. Control of fair division. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 67–73, 2016. URL: <http://www.ijcai.org/Abstract/16/017>.
- 3 Matthias Bentert, Robert Bredereck, Eva Deltl, Pallavi Jain, and Leon Kellerhals. How to resolve envy by adding goods. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.
- 4 Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 91–97, 2018. doi:10.24963/IJCAI.2018/13.
- 5 Niclas Boehmer, Robert Bredereck, Klaus Heeger, Dušan Knop, and Junjie Luo. Multivariate algorithmics for eliminating envy by donating goods. *Autonomous Agents and Multi-Agent Systems*, 38(2):43:1–43:35, 2024.
- 6 Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.
- 7 Felix Brandt and Anaëlle Wilczynski. On the convergence of swap dynamics to Pareto-optimal matchings. *Journal of Artificial Intelligence Research*, 80:1063–1098, 2024. doi:10.1613/JAIR.1.15305.
- 8 Robert Bredereck, Andrzej Kaczmarczyk, Junjie Luo, Rolf Niedermeier, and Florian Sachse. Improving resource allocations by sharing in pairs. *Journal of Artificial Intelligence Research*, 78:1069–1109, 2023. doi:10.1613/JAIR.1.15001.
- 9 Eric Budish, Gérard P. Cachon, Judd B. Kessler, and Abraham Othman. Course Match: a large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, 65(2):314–336, 2017. doi:10.1287/OPRE.2016.1544.

- 10 Harish Chandramouleeswaran, Prajakta Nimbhorkar, and Nidhi Rathi. Fair division in a variable setting. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 472–480, 2025. doi:10.5555/3709347.3743562.
- 11 Jiehua Chen, Joanna Kaczmarek, Paul Nüsken, Jörg Rothe, Ildikó Schlotter, and Tessa Seeger. Control in computational social choice. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.
- 12 Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. Reaching envy-free states in distributed negotiation settings. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1239–1244, 2007. URL: <http://ijcai.org/Proceedings/07/Papers/200.pdf>.
- 13 Anastasia Damamme, Aurélie Beynier, Yann Chevaleyre, and Nicolas Maudet. The power of swap deals in distributed resource allocation. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 625–633, 2015. URL: <http://dl.acm.org/citation.cfm?id=2773235>.
- 14 Britta Dorn, Ronald de Haan, and Ildikó Schlotter. Obtaining a proportional allocation by deleting items. *Algorithmica*, 83(5):1559–1603, 2021. doi:10.1007/S00453-020-00794-4.
- 15 Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7, pages 146–168. Cambridge University Press, 2016. doi:10.1017/CB09781107446984.008.
- 16 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.
- 17 Jonathan Goldman and Ariel D. Procaccia. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges*, 13(2):41–46, 2014. doi:10.1145/2728732.2728738.
- 18 Laurent Gourvès, Julien Lesca, and Anaëlle Wilczynski. Object allocation via swaps along a social network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 213–219, 2017. doi:10.24963/IJCAI.2017/31.
- 19 Jiatong Han and Warut Suksompong. Fast & Fair: a collaborative platform for fair division applications. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 23796–23798, 2024. doi:10.1609/AAAI.V38I21.30568.
- 20 Sen Huang and Mingyu Xiao. Object reachability via swaps under strict and weak preferences. *Autonomous Agents and Multi-Agent Systems*, 34(2):51:1–51:33, 2020.
- 21 Halvard Hummel and Magnus Lie Hetland. Maximin shares under cardinality constraints. In *Proceedings of the 19th European Conference on Multi-Agent Systems (EUMAS)*, pages 188–206, 2022. doi:10.1007/978-3-031-20614-6_11.
- 22 Ayumi Igarashi, Naoyuki Kamiyama, Warut Suksompong, and Sheung Man Yuen. Reachability of fair allocations via sequential exchanges. *Algorithmica*, 86(12):3653–3683, 2024. doi:10.1007/S00453-024-01271-Y.
- 23 Ayumi Igarashi and Tomohiko Yokoyama. Kajibuntan: a house chore division app. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, pages 16449–16451, 2023. doi:10.1609/AAAI.V37I13.27075.
- 24 Takehiro Ito, Yuni Iwamasa, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. Reforming an envy-free matching. *Algorithmica*, 87(4):594–620, 2025. doi:10.1007/S00453-025-01294-Z.
- 25 Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. On reachable assignments under dichotomous preferences. *Theoretical Computer Science*, 979:114196, 2023. doi:10.1016/J.TCS.2023.114196.
- 26 Fu Li, C. Gregory Plaxton, and Vaibhav B. Sinha. Object allocation over a network of objects: Mobile agents with strict preferences. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1578–1580, 2021. doi:10.5555/3463952.3464165.
- 27 Hervé Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003.

- 28 Hervé Moulin. Fair division in the internet age. *Annual Review of Economics*, 11:407–441, 2019.
- 29 Luis Müller and Matthias Bentert. On reachable assignments in cycles. In *Proceedings of the 7th International Conference on Algorithmic Decision Theory (ADT)*, pages 273–288, 2021. doi:10.1007/978-3-030-87756-9_18.
- 30 Jack Robertson and William Webb. *Cake-Cutting Algorithms: Be Fair if You Can*. Peters/CRC Press, 1998.
- 31 Erel Segal-Halevi. Redividing the cake. *Autonomous Agents and Multi-Agent Systems*, 36(1):14:1–14:36, 2022.
- 32 Hila Shoshan, Noam Hazon, and Erel Segal-Halevi. Efficient nearly-fair division with capacity constraints. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 206–214, 2023. doi:10.5555/3545946.3598638.
- 33 Xiaowei Wu, Bo Li, and Jiarui Gan. Approximate envy-freeness in indivisible resource allocation with budget constraints. *Information and Computation*, 303:105264, 2025. doi:10.1016/J.IC.2024.105264.
- 34 Sheung Man Yuen, Ayumi Igarashi, Naoyuki Kamiyama, and Warut Suksompong. Reforming an unfair allocation by exchanging goods. *CoRR*, abs/2412.19264, 2024. doi:10.48550/arXiv.2412.19264.