Fair Division with Allocator's Preference*

Xiaolin Bu[1], Zihao Li[2], Shengxin Liu[3], Jiaxin Song[1], and Biaoshuai Tao[1]

[1] Shanghai Jiao Tong University, Shanghai, China
{lin_bu,sjtu_xiaosong,bstao}@sjtu.edu.cn
[2] Nanyang Technological University, Singapore, Singapore
zihao004@e.ntu.edu.sg
[3] Harbin Institute of Technology, Shenzhen, China
sxliu@hit.edu.cn

Abstract. We consider the problem of fairly allocating indivisible resources to agents, which has been studied for years. Most previous work focuses on fairness and/or efficiency *among agents* given agents' preferences. However, besides the agents, the allocator as the resource owner may also be involved in many real-world scenarios (e.g., government resource allocation, heritage division, company personnel assignment, etc.). The allocator has the inclination to obtain a fair or efficient allocation based on her own preference over the items and to whom each item is allocated. In this paper, we propose a new model and focus on the following two problems concerning the allocator's fairness and efficiency:

1. Is it possible to find an allocation that is fair for both the agents and the allocator?
2. What is the complexity of maximizing the allocator's social welfare while satisfying the agents' fairness?

We consider the two fundamental fairness criteria: *envy-freeness* and *proportionality*. For the first problem, we study the existence of an allocation that is envy-free up to c goods (EF-c) or proportional up to c goods (PROP-c) from both the agents' and the allocator's perspectives, in which such an allocation is called *doubly EF-c* or *doubly PROP-c* respectively. When the allocator's utility depends exclusively on the items (but not to whom an item is allocated), we prove that a doubly EF-1 allocation always exists. For the general setting where the allocator has a preference over the items *and* to whom each item is allocated, we prove that a doubly EF-1 allocation always exists for two agents, a doubly PROP-2 allocation always exists for binary valuations, and a doubly PROP-$O(\log n)$ allocation always exists in general.

For the second problem, we provide various (in)approximability results in which the gaps between approximation and inapproximability ratios are asymptotically closed under most settings. When agents' valuations are binary, the problems of maximizing the social welfare in the allocator's perspective while ensuring agents' fairness criteria of PROP-c (with a general number of agents) and EF-c (with a constant number of agents) are both polynomial-time solvable for any positive integer c. For most of the other settings (general valuations, EF-c, etc.), we present strong inapproximability results.

Most of our results are based on some novel technical tools including the chromatic numbers of the Kneser graphs and linear-programming-based analysis.

Keywords: Fair Division · Allocator's Preference · EF-c/PORP-c.

1 Introduction

Fair division studies how to fairly allocate a set of resources to a set of agents with heterogeneous preferences. It is becoming a valuable instrument in solving real-world problems, e.g., Course Match for course allocation at the Wharton School in the University of Pennsylvania [16], and the website Spliddit (spliddit.org) for fair division of rent, goods, credit, and so on [23]. The construct of fair division was first articulated by Steinhaus [44,45] in the 1940s, and has become an attractive topic of interest in a wide range of fields, such as mathematics, economics, computer science, and so on (see, e.g., [13,41,40,38,3,47,2,32] for a survey).

The classic fair division problem mainly focuses on finding fair and/or efficient allocations *among agents given agents' preferences*. However, in many real-world scenarios, the allocator as the resource owner may also be involved, and, particularly, may have the inclination to obtain a fair or efficient allocation based on her own preference. For example, consider the division of inheritances, e.g., multiple companies and multiple

* A short version of this paper is accepted at the 19th Conference on Web and Internet Economics (WINE'23).

houses, from the parent to two children. Both children would prefer the companies as they believe the market value of the companies will be increased more than the houses in the future. At the same time, the parent may want to allocate the companies to the elder child since the parent thinks the elder child has a better ability to run the companies. The final allocation should be fair for children and may also need to incorporate the parent's ideas about the allocation. Another example is the government distributing educational resources (e.g., land, funding, experienced teachers or principals) among different schools. Some well-established schools may prefer land to build a new campus, while some new schools may need experienced teachers. On the other hand, the government may also have a preference (over the resources and to whom each resource is allocated) based on macroeconomic policy and may want the resulting distribution to be efficient on top of each school feels that it gets a fair share. Other examples abound: a company allocates resources to multiple departments, an advisor allocates tasks/projects to students, a conference reviewer assignment system allocates papers to reviewers, etc.

We focus on the allocation of indivisible goods in this work. To measure fairness, the two most fundamental criteria in the literature are *envy-freeness* and *proportionality*, respectively [44,45,21,48]. In particular, an allocation is said to be envy-free if each agent weakly prefers her bundle over any other agent's based on her own preference, and proportional if each agent values her bundle at least $1/n$ of her value for the whole resources, where n is the number of agents. Both fairness criteria can always be achieved in divisible resource allocation but it is not the case for indivisible resources (say, a simple example with two agents and one good). This triggers an increasing number of research work to consider relaxing exact fairness notions of envy-freeness and proportionality to *envy-freeness up to c goods (EF-c)* and *proportionality up to c goods (PROP-c)* (see, e.g., [31,15,19]). Specifically, an allocation is said to be EF-c if any agent's envy towards another agent could be eliminated by (hypothetically) removing at most c goods in the latter's bundle, and PROP-c if any agent's fair share of $1/n$ could be guaranteed by (hypothetically) adding at most c goods that are allocated to other agents, where c is a positive integer. Besides fairness, another important issue of fair division is *(economic) efficiency* (e.g., social welfare), which is used to measure the total happiness of the agents [18,12,9,4].

The fair division problem with allocator's preference presents new challenges compared to the classic fair division problems. With indivisible goods, it is well known that the *round-robin algorithm* [4] can return a fair, i.e., EF-1, allocation from the agents' perspective. However, this algorithm cannot be easily adapted to the problem where both agents and the allocator have preferences over items. Specifically, an agent's preference describes how much this agent values each item, while the allocator's preference describes how much the allocator regards each item values for each agent. Consider the instance with both agents' and the allocator's preferences shown in Tables 1 and 2.

	Item 1	Item 2	Item 3
Agent 1	2	1	0
Agent 2	0	1	2

Table 1. Agents' Preferences

	Item 1	Item 2	Item 3
Allocator for Agent 1	0	2	1
Allocator for Agent 2	1	2	0

Table 2. Allocator's Preferences

Suppose, without loss of generality, agent 1 is before agent 2 in the ordering of the round-robin algorithm. When performing the algorithm without considering the allocator's preference, agent 1 gets a bundle of items 1 and 2 while agent 2 gets item 3. From the allocator's perspective, this allocation is not EF-1 since the allocator thinks agent 2 will envy agent 1 even when an arbitrary item is removed from agent 1's bundle. One can also verify that the above allocation is not social welfare maximizing based on the allocator's viewpoint, i.e., the allocator thinks there is another allocation such that the total happiness of the agents is larger. On the other hand, performing the round-robin algorithm based solely on the allocator's preference will return an allocation where agent 1 gets items 2 and 3 while agent 2 gets item 1 (assuming agent 1 has a higher priority in the ordering). This allocation is not EF-1 from the agents' perspective as the envy from agent 2 to agent 1 cannot be eliminated by removing a single item in agent 1's bundle. Thus, it is tempting to ask: How to find fair or efficient allocations in the presence of agents' and the allocator's preferences? Specifically, we want to answer the following two questions in this paper.

[4] The round-robin algorithm works as follows: Given an ordering of agents, each agent picks her favorite item among the remaining items to her bundle following the ordering in rounds until there is no remaining item.

n	**Fairness**	v_i	u_i	**Negative Results**	**Positive Results**
2	EF-c	arbitrary	arbitrary	2 (Thm 7)	2 (Thm 8)
	EF-c	arbitrary	binary	2 (Thm 7)	2 (Thm 8)
	EF-c	binary	arbitrary	—	1 (Thm 10)
constant	EF-c	arbitrary	binary	$\left\lfloor \frac{1+\sqrt{4n-3}}{2} \right\rfloor$ [14]	?
	EF-c	binary	arbitrary	—	1 (Thm 10)
general	EF-c	binary	binary	$m^{1-\epsilon}, n^{1/2-\epsilon}$ (Thm 11)	m (Thm 12)
	EF-c	arbitrary	arbitrary	$m^{1-\epsilon}, n^{1/2-\epsilon}$ (Thm 11)	m (Thm 12)
	PROP-c	arbitrary	binary	2 (Thm 13)	?
	PROP-c	binary	arbitrary	—	1 (Thm 14)

Table 3. Positive and Negative Results of Maximizing Allocator's Efficiency. The numbers of agents and items are denoted by n and m, respectively. For each agent i, v_i represents her utility function while u_i represents how much the allocator regards each item values for agent i. Numbers α for negative results indicate that the problem is NP-hard to approximate to within the ratio α; numbers α for positive results indicate that the problem admits a polynomial time α-approximation algorithm. All our negative results hold for $c = 1$.

Question 1: Is it possible to find an allocation that guarantees both the allocator's and agents' fairness?
Question 2: What is the complexity of maximizing the allocator's efficiency while ensuring agents' fairness?

1.1 Our Results

We initiate the study of fair division with allocator's preference and address the two research questions above in this paper. We mainly focus on the allocation of *indivisible* resources and discuss the *divisible* resources in the appendix (which also includes omitted proofs in the paper).

For the first problem, we propose new fairness notions *doubly EF-c* and *doubly PROP-c* that extend EF-c and PROP-c to our setting with regard to the allocator's preference. We first consider the setting where the allocator's utility only depends on the items (but not to whom an item is allocated), and we show that a doubly EF-1 allocation always exists. We then consider the general setting where the allocator's utility depends on both the items and the allocation. For two agents, we show that 1) a doubly EF-1 allocation always exists, and 2) a doubly EF-2 allocation and a doubly PROP-1 allocation can be computed in polynomial time. For a general number of agents, we show that a doubly PROP-$\log_2 n$ allocation always exists for n being an integer power of 2, and we show that a doubly PROP-$(2\lceil \log n \rceil)$ allocation always exists and can be computed in polynomial time. If we restrict to binary valuations, we show that a doubly PROP-2 allocation always exists and can be computed in polynomial time.

For the second problem, we study its complexity and approximability for both binary and general (additive) valuations. Our results are presented in Table 3. The gap between the approximation ratio and the inapproximability ratio is closed, or asymptotically closed, under most settings. If agents' valuations are binary, this problem is tractable for EF-c with a constant number of agents and for PROP-c with a general number of agents. Under most other settings, this problem admits strong inapproximability ratios even for $c = 1$.

Our results use many technical tools that are uncommon in the fair division literature, including i) the chromatic numbers of generalized Kneser graphs and ii) some linear programming-based analyses.

For i), we use a generalized Kneser graph to model a set of allocations and the relations between the allocations. Specifically, the set of allocations that are not fair based on an agent's valuation form an independent set in the graph. The existence of a doubly fair allocation is built upon the argument that there are still remaining vertices after removing all vertices that correspond to unfair allocations. Since the set of unfair allocations for each agent forms an independent set, the chromatic number of the graph plays an important role in our analysis.

For ii), we use linear programs to formulate our problems. The solution to the linear program naturally corresponds to a *fractional* allocation. Our technique is mainly based on the analysis of the vertices of the polytope defined by the linear program. In some applications, we bound the number of the fractional items in an allocation given by a vertex solution of the linear program, and then handle those few fractional items separately. In other applications, we prove that all the vertex solutions of the linear program are integral.

1.2 Further Related Work

Conceptually, our model with allocator's preference shares similarities with recent research work on fair division with two-sided fairness, e.g., [39,24,22,28]. The existing two-sided fairness literature studies the fair division problem where there are two disjoint groups of agents and each agent in one group has a preference over the agents of the other group. The objective is then to find a (many-to-many) matching that is fair to each agent with respect to her belonging group. We remark these two models are different due to the following major reasons:

- In their model, there are two disjoint sets of agents, and each group of preferences is defined from one set of agents to the other set of agents (viewed as a set of "goods"). On the other hand, the two groups of preferences (one is from the agents and the other one is from the allocator) in our setting are both defined on a single set of agents and a single set of goods.
- In their model, each agent will be allocated (or matched) a set of agents from the other group which is different from ours, whereas the allocator in our model will not receive any resource in the allocation.

As we can see, our model with allocator's preference reduces to the standard setting of indivisible goods when the allocator's preference coincides with agents' preferences. Our first research question reduces to find EF-c or PROP-c allocations in indivisible fair allocation, where the fairness notions of EF-1 and PROP-1 are are extensively studied. In particular, an EF-1 allocation always exits and can be computed in polynomial time [31,17]. For PROP-1, an allocation that is PROP1 and Pareto optimal always exits and can be computed in polynomial time [19,10,7,37]. When considering the issue of economic efficiency, the problem in our second research question could be mapped to the problem of maximizing social welfare within either EF-1 or PROP-1 allocations in the indivisible goods setting. Aziz et al. [4] showed that the problem with either the EF-1 or the PROP-1 condition is NP-hard for $n \geq 2$ and Barman et al. [9] showed that the problem with the EF-1 requirement is NP-hard to approximate to within a factor of $1/m^{1-\varepsilon}$ for any $\varepsilon > 0$ for general numbers of agents n and items m. Later, Bu et al. [14] gave a complete landscape for the approximability of the problem with the EF-1 criterion.

In addition, several works studied the fair division problem where the resources need to be allocated among *groups* of agents and the resources are shared among the agents within each predefined group [34,42,46,43]. In their model, $n = n_1 + \cdots + n_k$ agents will be divided into $k \geq 2$ groups, where group i contains $n_i \geq 1$ agents. An allocation is a partition of goods into k groups. Each agent in the i-th group extracts utilities according to the i-th bundle. Kyropoulou et al. [30] also generalized the classic EF-c to the group setting: An agent's envy towards another group could be eliminated by removing at most c goods from that group's bundle. PROP-c could be defined similarly ([35]). With binary valuations, Kyropoulou et al. [30] gave the characterization of the cardinalities of the groups for which a group EF-1 allocation always exits. In particular, they showed that a group EF-1 allocation always exists when there are two groups and each group contains two agents with binary valuations. Subsequently, Manurangsi and Suksompong [35] showed via the discrepancy theory that EF-$O(\sqrt{n})$ and PROP-$O(\sqrt{n})$ allocations always exist in the group setting. Note that, when each group contains exactly two agents, i.e., $n_1 = \ldots = n_k = 2$, the fair division problem in the predefined group setting coincides with our model (where each group could be considered to have an agent and the allocator). However, we obtain improved results in this particular setting through different technical tools.

2 Preliminaries

Let $[k] = \{1, \ldots, k\}$. Our model consists of a set of agents $N = [n]$, a set of indivisible items $M = \{g_1, \ldots, g_m\}$, and *the allocator*. Each agent i has a nonnegative *utility function* $v_i : \{0,1\}^m \to \mathbb{R}_{\geq 0}$. In addition, the allocator has her own preference in our model. The allocator's preference is composed by n utility functions $u_i : \{0,1\}^m \to \mathbb{R}_{\geq 0}$ where each u_i is used to describe how much the allocator regards each item values for agent i.

We assume both utility functions u_i and v_i are *additive*, which means $v_i(X) = \sum_{g \in X} v_i(g)$ and $u_i(X) = \sum_{g \in X} u_i(g)$ for any bundle $X \subseteq M$. A utility function v_i (or u_i) is said to be *binary* if $v_i(g) \in \{0,1\}$ for any item $g \in M$. An *allocation* of the items $\mathcal{A} = (A_1, A_2, \ldots, A_n)$ is an ordered partition of M, where A_i is the bundle of items allocated to agent i.

Now we are ready to introduce the fairness notions we consider. Let c be a nonnegtive integer.

Definition 1 (Envy-free up to c goods). *An allocation $\mathcal{A}$ is said to be envy-free up to c goods (EF-c) if for all pairs of agents $i \neq j$, there exists a set $B \subseteq A_j$ such that $|B| \leq c$ and $v_i(A_i) \geq v_i(A_j \setminus B)$ (or $v_i(A_i) \geq v_i(A_j) - v_i(B)$ for additive utility functions).*

Definition 2 (Proportional up to c goods). *An allocation $\mathcal{A}$ is said to be proportional up to c goods (PROP-c) if for any agent i, there exists a set $B \subseteq M \setminus A_i$ such that $|B| \leq c$ and $v_i(A_i \cup B) \geq \frac{1}{n} v_i(M)$ (or $v_i(A_i) \geq \frac{1}{n} v_i(M) - v_i(B)$ for additive utility functions).*

Clearly, EF-c implies PROP-c for additive utility functions. It is also well known that an EF-1 (hence, PROP-1) allocation always exists and can be computed in polynomial time [31,17].

In our model, besides ensuring fairness among agents, we also consider allocator's fairness. Thus, we generalize the above fairness criteria in the following.

Definition 3 (Doubly Envy-free up to c goods). *An allocation $\mathcal{A}$ is said to be doubly envy-free up to c goods (Doubly EF-c) if for all pairs of agents $i \neq j$, there exist sets $B_1, B_2 \subseteq A_j$ such that $|B_1|, |B_2| \leq c$, $v_i(A_i) \geq v_i(A_j \setminus B_1)$ and $u_i(A_i) \geq u_i(A_j \setminus B_2)$.*

Definition 4 (Doubly Proportional up to c goods). *An allocation $\mathcal{A}$ is said to be doubly proportional up to c goods (Doubly PROP-c) if for any $i \in N$, there exist sets $B_1, B_2 \subseteq M \setminus A_i$ such that $|B_1|, |B_2| \leq c$, and $v_i(A_i \cup B_1) \geq \frac{1}{n} v_i(M)$, and $u_i(A_i \cup B_2) \geq \frac{1}{n} u_i(M)$.*

When the allocator's utility functions are identical to agents' utility functions, it is easy to see that doubly EF-c and doubly PROP-c degenerate to EF-c and PROP-c respectively. The above defined double fairness notions with the allocator's preference can also be interpreted as: There are two groups of valuation functions u and v where one is from the agents and the other one is from the allocator. A single allocation is said to satisfy double fairness if such an allocation is fair, e.g., doubly EF-c/PROP-c, with respect to both valuation functions u and v.

To measure the economic efficiency for the allocator, we consider *allocator's efficiency*:

Definition 5. *The* allocator's efficiency *of an allocation $\mathcal{A} = (A_1, \ldots, A_n)$, denoted by* $\texttt{EFFICIENCY}(\mathcal{A})$*, is the summation of the allocator's utilities of all the agents* $\texttt{EFFICIENCY}(\mathcal{A}) = \sum_{i=1}^{n} u_i(A_i)$.

In this paper, we are interested in the following two problems.

Problem 1. Given a set of indivisible items M, a set of agents $N = [n]$ with their utility functions $(v_1, \ldots, v_n)$, and the allocator with her preference $(u_1, \ldots, u_n)$, determine whether there exists an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ that is doubly EF-c/PROP-c.

Problem 2. Given a set of indivisible items M, a set of agents $N = [n]$ with their utility functions $(v_1, \ldots, v_n)$, and the allocator with her preference $(u_1, \ldots, u_n)$, the problem of *maximizing allocator's efficiency subject to EF-c/PROP-c* aims to find an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ that maximizes allocator's efficiency $\texttt{EFFICIENCY}$ subject to that $\mathcal{A}$ is EF-c/PROP-c.

2.1 Kneser Graph and Chromatic Number

Let n, k be two integers. The Kneser graph $\mathcal{K}(n, k)$ is the graph with the set of all the k-element subsets of $[n]$ as its vertex set and two vertices are adjacent if their intersection is empty. It was further extended to the following generalized version. Given three integers $n, k, s \in \mathbb{Z}^+$, in the generalized Kneser graph $\mathcal{K}(n, k, s)$, two vertices are adjacent if and only if their corresponding subsets intersect in s or fewer elements.

The *chromatic number* of a graph is the minimum number of colors needed to color the vertices such that no two adjacent vertices have the same color. In other words, the vertices with the same color form an independent set. We denote the chromatic number of a kneser graph $\mathcal{K}(n, k, s)$ by $\chi(n, k, s)$. For instance, when $n = 4, k = 3, s = 2$, the kneser graph has $\binom{4}{3} = 4$ vertices and every two vertices are adjacent. Thus, $\mathcal{K}(4, 3, 2)$ is a clique and $\chi(4, 3, 2) = 4$.

For the chromatic number of the Kenser graph, when $n \geq 2k$, it is exactly equal to $n - 2k + 2$ [33,25,8,36]. For the generalized Kneser graph, Jafari and Moghaddamzadeh [29] gave the following lower bound:

Lemma 1 **([29]).** *For any positive integers $s < k < n$,*

$$\chi(n, k, s) \geq n - 2k + 2s + 2.$$

As a special case, they gave the exact chromatic number when $n = 2k$, as follows:

Lemma 2 **([29]).** *For any $k \in \mathbb{Z}^+ \geq 2$, $\chi(2k, k, 1) = 6$.*

2.2 Totally Unimodular Matrix and Linear Programming

Totally unimodular matrix is a special family of matrices which can be used to check whether a linear programming is *integral*, i.e., there exists one optimal solution such that all decision variables are integers.

Definition 6 (Totally Unimodular Matrix). *A matrix $\mathbf{A}_{m\times n}$ is a totally unimodular matrix (TUM) if every square submatrix of $\mathbf{A}$ has determinant 0, +1 or −1.*

To determine whether a matrix is TUM, we have the following lemma:

Lemma 3. *Given a matrix $\mathbf{A} \in \{0, \pm 1\}^{m\times n}$, $\mathbf{A}$ is TUM if it can be written as the form of $\begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}$, where $\mathbf{A}_1 \in \{0, 1\}^{r\times n}$ (or $\{0, -1\}^{r\times n}$), $\mathbf{A}_2 \in \{0, 1\}^{(m-r)\times n}$ (or $\{0, -1\}^{(m-r)\times n}$), $1 \leq r \leq m$ and there is at most one nonzero number in every column of $\mathbf{A}_1$ or $\mathbf{A}_2$.*

Proof. We prove this by induction. Assume the square submatrix $\mathbf{A}'$ of $\mathbf{A}$ is an $n' \times n'$ matrix. It holds for the case when $n' = 1$ since all entries are 0, −1 or 1. We next assume $n' > 1$. If there exists one column with only one non-zero entry, and we assume the square submatrix after removing the corresponding row and column of this entry is $\mathbf{B}$, we have $\det(\mathbf{A}) = \pm\det(\mathbf{B})$, by induction, $\det(\mathbf{A}')$ is also equal to 0, −1 or 1.

If there is no such column, since there are at most two non-zero entries in one column of the original matrix $\mathbf{A}$, there are exactly two entries in each column. Then, we consider the following linear combination of rows in $\mathbf{A}'$. If $\mathbf{A}_1$ and $\mathbf{A}_2$ consist the same non-zero values, we add all rows in $\mathbf{A}_1$ and minus all rows in $\mathbf{A}_2$. Otherwise, we add all rows in both $\mathbf{A}_1$ and $\mathbf{A}_2$. The above linear combination equal to a zero vector, which implies $\det(\mathbf{A}') = 0$. Thus, $\mathbf{A}$ is totally unimodular. □

Lemma 4 **([27]).** *If $\mathbf{A}$ is totally unimodular and $\mathbf{b}$ is an integer vector, then each vertex of the polytope $\{\mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ has integer coordinates.*

We can further show there exist polynomial-time algorithms to find the optimal vertex solution for such a linear program by the following lemma.

Lemma 5 **([26]).** *For a linear program $\max\{\mathbf{c}^\top\mathbf{x} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, if optimal solutions exist, an optimal vertex solution can be found in polynomial time. In particular, we can find an (integral) vertex of the polytope $\{\mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ in polynomial time.*

3 Double Fairness

In this part, we present the results for the existence of double fairness. In the first part, we assume the allocator's utility depends exclusively on the item (rather than to whom an item is allocated). That is, we assume the allocator's utility functions are identical $u_1 = \cdots = u_n$. We show that a doubly EF-1 allocation always exists in this case by adapting the envy-cycle procedure. After that, we consider the general setting without $u_1 = \cdots = u_n$. In this case, we first show that a doubly EF-1 allocation always exists for $n = 2$ based on the chromatic number of the generalized Kneser graph $\mathcal{K}(m, m/2, 1)$. However, the existence of doubly EF-1 allocations for $n > 3$ is highly non-trivial. For this reason, we consider relaxing the fairness constraint to doubly EF-c or PROP-c and try to minimize the value of c. We show that a doubly PROP-$O(\log n)$ allocation always exists for any number of agents via the techniques based on the generalized Kneser graph and linear programming. Finally, we also consider another common setting, where both the allocator and agents' utility functions are binary (the utility value can only be 0 or 1). This relaxation makes the problem tractable and we demonstrate a doubly PROP-2 allocation always exists in this setting.

Algorithm 1: Algorithm to Find Doubly EF-1 Allocation for Identical Utility Function

Input: the set of agents N, the set of items M, agents' utility functions v_i, allocator's utility function u
Output: a doubly EF-1 allocation

Let $G = (V, E)$ be the envy-graph where each vertex represents an agent and $E \leftarrow \emptyset$;
Initialize $\mathcal{A} = (\emptyset, \dots, \emptyset)$;
if $n \nmid m$ **then**
  Add dummy items to M such that $n \mid m$ and set the utility of each dummy item as 0;
Let M_s be the sorted array of the items according to allocator's utility function u in descending order;
for *every* n *items* $M_n \subseteq M_s$ **do**
  Let $\{i_1, \dots, i_n\}$ be the agents in topological order of graph G;
  for *each* $j \in \{1, \dots, n\}$ **do**
    Allocate agent i_j's favorite item $g \in M_n$ to i_j: $A_{i_j} \leftarrow A_{i_j} \cup \{\text{argmax}_{g \in M_n} v_{i_j}(g)\}$;
    $M_n \leftarrow M_n \setminus \{g\}$;
  Update the envy-graph G;
  Iteratively run the cycle-elimination algorithm and update G until G contains no cycle;
Remove the dummy items from the allocation $\mathcal{A}$;
return *the allocation* $\mathcal{A}$

3.1 Identical Allocator's Utility Function

This section considers the case when the allocator's utility functions $u_1, \dots, u_n$ are identical. Let $u = u_1 = \dots = u_n$.

We first give a brief introduction of the techniques used in this section. The envy-cycle procedure was first proposed by [31] to compute an EF-1 allocation for general valuations. In the envy-cycle procedure, an *envy-graph* is constructed for a partial allocation. Each vertex in the envy-graph represents an agent and each directed edge (u, v) means that agent u envies agent v in the current allocation. When there is a cycle in an envy-graph, we use the cycle-elimination algorithm to eliminate this cycle.

Definition 7 (Cycle-elimination Algorithm). *Given an envy-graph with a cycle $u_1 \to \dots \to u_n \to u_1$, shift the agents' bundles along the cycle ($A_{u_i} \leftarrow A_{u_{i+1}}$ for $i = 1, \dots, n-1$ and $A_n \leftarrow A_1$).*

Theorem 1. *When the allocator's utility functions are identical, a doubly EF-1 allocation always exists for any number of agents n, and can be found by Algorithm 1 in polynomial time.*

Before we prove Theorem 1, we first describe our algorithm. At the beginning of the algorithm, we construct an envy-graph G with n vertices and no edges and sort the items according to the allocator's utility function in descending order. Then, we divide the sorted items into $\lceil \frac{m}{n} \rceil$ groups where each group contains n items. In each round, we allocate a group of items to the agents such that each agent receives exactly one item. In particular, each agent takes away her favorite item from the group, where the agents are sorted in the topological order of G before the iteration begins. After all these n items are allocated, we update the envy-graph and run the cycle-elimination algorithm, so that the envy-graph contains no cycle and a topological order of the agents can be successfully found in the next round.

To prove Theorem 1, we first prove the allocation is EF-1 from both the agents' and the allocator's perspectives.

Lemma 6. *The allocation computed by Algorithm 1 is EF-1 to the agents.*

Proof. We use induction to show that EF-1 is maintained to the agents through the algorithm. At the beginning of the first round, the allocation is empty, so it is EF-1. We assume at the beginning of the ℓ-th round, the allocation $\mathcal{A}$ is EF-1. Denote the allocation after running the $(\ell+1)$-th round by $\mathcal{B}$. We now show $\mathcal{B}$ is still EF-1.

We first consider the allocation $\mathcal{A}'$ before the cycle-elimination algorithm. Consider two arbitrary agents i, j, and assume they receive items g_i and g_j respectively in the $(\ell+1)$-th round. Without loss of generality, we assume i is before j in the topological order of G after the ℓ-th round. For agent i, since $\mathcal{A}$ is EF-1, there exists an item $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$. Since i is before j, $v_i(g_i) \geq v_i(g_j)$. have

$$v_i(A_i') = v_i(A_i \cup \{g_i\}) \geq v_i((A_j \setminus \{g\}) \cup \{g_j\}) = v_i(A_j' \setminus \{g\}),$$

so agent i will not envy agent j if g is removed from A_j'. For agent j, she does not envy i in $\mathcal{A}$, so $v_j(A_j) \geq v_j(A_i)$. Then, we have

$$v_j\left(A_j'\right) = v_j(A_j \cup \{g_j\}) \geq v_j(A_i) = v_j(A_i' \setminus \{g_i\}),$$

so j will not envy i if g_i is removed from A_i'. Hence, $\mathcal{A}'$ is an EF-1 allocation to the agents.

The cycle-elimination algorithm does not destroy the EF-1 property. The allocation $\mathcal{B}$ after cycle-elimination is a permutation of $\mathcal{A}'$ where the constituents of each bundle do not change, and each agent receives a bundle with a weakly higher value. Hence, $\mathcal{B}$ is still EF-1 to the agents. □

Lemma 7. *The allocation found by Algorithm 1 is EF-1 to the allocator.*

Proof. We prove this by induction. At the beginning of the algorithm, the empty allocation is EF-1 to the allocator, and we assume at the beginning of the ℓ-th round, the allocation is EF-1. We now prove after the ℓ-th round, the allocation is still EF-1. Let $x^{(k)}$ represent the item added into bundle X in the k-th round.

Consider two arbitrary bundles X, Y, and assume two items add to these bundles are g_i and g_j respectively. Without loss of generality, we assume $u(g_i) \geq u(g_j)$. Suppose the two bundles are updated to X', Y' after running the ℓ-th round. For bundle X, since there exists an item $g \in Y$ such that $u(X) \geq u(Y \setminus \{g\})$, we have

$$u(X') = u(X \cup \{g_i\}) \geq u(Y \cup \{g_j\} \setminus \{g\}) = u(Y' \setminus \{g\}).$$

For bundle Y, because the items are sorted in descending order, we have $u\left(y^{(k-1)}\right) \geq u\left(x^{(k)}\right)$ for $2 \leq k \leq \ell - 1$, and $u\left(y^{(\ell-1)}\right) \geq u(g_i)$. Then we have

$$u(Y') \geq \sum_{k=2}^{\ell} u\left(y^{(k-1)}\right) \geq \sum_{k=2}^{\ell-1} u\left(x^{(k)}\right) + u(g_i) = u\left(X' \setminus \{x^{(1)}\}\right).$$

Hence, the allocation after the ℓ-th round is still EF-1 to the allocator. □

We conclude from the above two lemmas that the output allocation is doubly EF-1. Moreover, sorting the items takes $O(m \log m)$ time. To allocate each group of items, finding a topological order of G costs $O(n^2)$, and allocating one item and updating the envy-graph cost $O(n)$. The cycle-elimination algorithm takes $O(n^2)$ time to find a cycle and runs for at most $O(n^2)$ iterations because at least one edge is eliminated in each iteration. This process repeats for $\lceil m/n \rceil$ rounds. The overall complexity of Algorithm 1 is $O\left(m\left(\log m + n^3\right)\right)$. Hence, Theorem 1 holds.

3.2 General Additive Valuations with Two Agents

For general (monotone) valuations with two agents, the existence of a doubly EF-1 allocation can be proved with the help of the generalized Kneser graph.

Theorem 2. *When $n = 2$, there always exists a doubly EF-1 allocation.*

Proof. Our high-level idea is to consider the set of allocations that some agents or the allocator does not regard as EF-1, then we use the Kneser graph to show the union of these four sets for the four utility functions v_1, v_2, u_1, u_2 cannot cover all the allocations. We assume the number of items m is even. Otherwise, we can add a dummy item g such that $v_i(g) = u_i(g) = 0$ for $i = 1, 2$. It needs to be mentioned that, in the following proof, we only consider the allocations where each bundle's size is exactly $\frac{m}{2}$. We denote the set of such allocations by Π.

Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be the set of allocations that the two agents respectively do not regard as EF-1. Besides, $\mathcal{U}_1$ and $\mathcal{U}_2$ are respectively the set of allocations that the allocator does not regard as EF-1 for agent 1 and agent 2. Formally, they are given by the following definitions:

$$\mathcal{V}_1 \triangleq \{\mathcal{A} \in \Pi : v_1(A_1) < v_1(A_2 \setminus \{g\}), \forall g \in A_2\}, \mathcal{V}_2 \triangleq \{\mathcal{A} \in \Pi : v_2(A_2) < v_2(A_1 \setminus \{g\}), \forall g \in A_1\},$$
$$\mathcal{U}_1 \triangleq \{\mathcal{A} \in \Pi : u_1(A_1) < u_1(A_2 \setminus \{g\}), \forall g \in A_2\}, \mathcal{U}_2 \triangleq \{\mathcal{A} \in \Pi : u_2(A_2) < u_2(A_1 \setminus \{g\}), \forall g \in A_1\}.$$

To show the existence of doubly EF1 allocation, it suffices to show that $\mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{U}_1 \cup \mathcal{U}_2 \subsetneq \Pi$.

Let $\mathcal{V}_1^{(1)} = \{A_1 : (A_1, A_2) \in \mathcal{V}_1\}$, and define $\mathcal{V}_2^{(1)}, \mathcal{U}_1^{(1)}, \mathcal{U}_2^{(1)}$ analogously. We give the following proposition for $\mathcal{V}_1^{(1)}$, which also works for the other three sets.

Proposition 1. *For each $A_1, A_1' \in \mathcal{V}_1^{(1)}$, $|A_1 \cap A_1'| \geq 2$.*

Proof. For the sake of contradiction, we assume $|A_1 \cap A_1'| \leq 1$. If $A_1 \cap A_1' = \emptyset$, (A_1, A_1') is an allocation. If (A_1, A_1') is not EF1 according to v_1, then (A_1', A_1) is envy-free, which means $A_1' \notin \mathcal{V}_1^{(1)}$.

If $|A_1 \cap A_1'| = 1$, let g_1 be the item in $A_1 \cap A_1'$ and g_2 be the only item in $M \setminus (A_1 \cup A_1')$. According to the definition of $\mathcal{V}_1$, we have

$$v_1(A_1) < v_1(M \setminus A_1) - v_1(g_2) = v_1(A_1') - v_1(g_1), v_1(A_1') < v_1(M \setminus A_1') - v_1(g_2) = v_1(A_1) - v_1(g_1).$$

Combining the above two inequalities yields a contradiction. □

Return to the proof of Theorem 2. We consider the generalized Kneser graph $\mathcal{H} = \mathcal{K}\left(m, \frac{m}{2}, 1\right)$. Each vertex of the graph defines a bundle B of $m/2$ items, and it defines an allocation (A_1, A_2) where $A_1 = B$ and $A_2 = M \setminus B$. Due to Proposition 1, each of $\mathcal{V}_1^{(1)}, \mathcal{V}_2^{(1)}, \mathcal{U}_1^{(1)}, \mathcal{U}_2^{(1)}$ cannot contain two adjacent vertices of $\mathcal{H}$ and is thus an independent set.

Finally, we have $\mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{U}_1 \cup \mathcal{U}_2 \subsetneq \Pi$. Otherwise, $\mathcal{H}$ can be decomposed into four independent sets, which contradicts to $\chi(\mathcal{H}) = 6$ (Lemma 2). □

Remark 1. Theorem 2 also holds for general monotone utility functions that are not necessarily additive, with the same proof above.

Theorem 2 is non-constructive. For constructive results, we use linear programming to construct a doubly EF-2 allocation in Theorem 3. Since the techniques used in this theorem are also used in the proof of Lemma 9, we defer the proof of the following theorem to after Lemma 9.

Theorem 3. *When $n = 2$, there always exists a doubly EF-2 allocation that can be computed in polynomial time.*

It is easy to see that an EF-2 allocation is always PROP-1 for two agents.

Corollary 1. *When $n = 2$, there always exists a doubly PROP-1 allocation that can be computed in polynomial time.*

3.3 General Additive Valuations with General Number of Agents

Next, we consider the lower bound of c when $n \geq 2$. Before presenting the details of the following two proofs, here we give our high-level ideas. We refer to the idea of Even-Paz algorithm [20]. Given n agents, we first partition the agent set into two groups and try to allocate each group one bundle. After that, we fix the two bundles to the two groups and then do further allocating within groups recursively. To guarantee the property of proportionality, it is needed to ensure that each agent and the allocator regard the ratio of the value the agent's group receives as about $\frac{1}{2}$.

For the clarity of exposition, we give the following definition (used only for technical purposes).

Definition 8. *Given a 2-partition of the agent set (N_1, N_2) and two integer $k_1, k_2 \in \mathbb{Z}^+$. We say that (X_1, X_2) is a 2-*balanced PROP-(k_1, k_2) allocation with respect to (N_1, N_2) *if $v(X_1) \geq \frac{|N_1|}{n} v(M) - L(v, k_1, X_2)$ for each agent $i \in N_1, v \in \{v_i, u_i\}$ and $v(X_2) \geq \frac{|N_2|}{n} v(M) - L(v, k_2, X_1)$ for each agent $i \in N_2$ and $v \in \{v_i, u_i\}$, where $L(v, t, S) \in \mathbb{R}_{\geq 0}$ is the sum of the values of the $\min\{t, |S|\}$ items in S with the largest values under the function v.*

We give two lemmas of the existence of 2-balanced PROP-(k_1, k_2) allocations. In the first one, we show that, for n being an even number, there always exists a 2-balanced PROP-$\left(\frac{n}{2}, \frac{n}{2}\right)$ allocation via Kneser graph. In the second one, we give a constructive proof of how to find a 2-balanced PROP-$(n-1, n)$ allocation via linear programming, and it does not require n being even.

Lemma 8. *If n is even, then for any 2-partition (N_1, N_2) such that $|N_1| = |N_2| = \frac{n}{2}$, there always exists a 2-balanced PROP-$\left(\frac{n}{2}, \frac{n}{2}\right)$ allocation.*

Proof. Let $n = 2s$. For each agent $i \in N_1$, we enumerate the bundles that she still does not regard as proportional even if further taking s largest items from the remaining items, as follows:

$$\mathcal{V}_i \triangleq \left\{X \in \mathit{\Pi} : v_i(X) < \frac{v_i(M)}{2} - L\left(v_i, s, M \setminus X\right)\right\}.$$

Inversely, for each agent $i \in N_2$, we enumerate the bundles that she regards as proportional when taking s largest items from the remaining items,

$$\mathcal{V}_i \triangleq \left\{X \in \mathit{\Pi} : v_i(X) > \frac{v_i(M)}{2} - L\left(v_i, s, X\right)\right\}.$$

By replacing v_i by u_i in the above definition, we similarly define $\mathcal{U}_i \subseteq M$ for each agent $i \in N$.

Proposition 2. *For each agent $i \in N_1$, $|X_1 \cap X_2| > n$ for any $X_1, X_2 \in \mathcal{V}_i$.*

Proof. For the sake of contradiction, we assume $|X_1 \cap X_2| \leq n$. Let $B = M \setminus (X_1 \cup X_2)$. Hence, $|B| \leq m - 2 \cdot \frac{m}{2} + n \leq n$. According to the definition of $\mathcal{V}_i$, we have

$$v_i(X_1) < \frac{1}{2}v_i(M) - L\left(v_i, s, M \setminus X_1\right), v_i(X_2) < \frac{1}{2}v_i(M) - L\left(v_i, s, M \setminus X_2\right)$$

Sum them up, and we have

$$v_i(X_1) + v_i(X_2) < v_i\left(X_1 \cup X_2\right) + v_i(B) - L\left(v_i, s, M \setminus X_1\right) - L\left(v_i, s, M \setminus X_2\right).$$

Since $B = M \setminus (X_1 \cup X_2) \subseteq M \setminus X_1$, then $L(v_i, s, B) \leq L(v_i, s, M \setminus X_1)$. For the same reason, $L(v_i, s, B) \leq L(v_i, s, M \setminus X_2)$. Thus,

$$v_i(X_1) + v_i(X_2) < v_i\left(X_1 \cup X_2\right) + v_i(B) - L\left(v_i, s, B\right) - L\left(v_i, s, B\right) < v_i\left(X_1 \cup X_2\right),$$

which leads to a contradiction and concludes this proposition. □

Proposition 3. *For each agent $i \in N_2$, $|X_1 \cap X_2| > n$ for any $X_1, X_2 \in \mathcal{V}_i$.*

Proof. For the sake of contradiction, we assume $|X_1 \cap X_2| \leq n$. Hence,

$$\begin{aligned}
v_i(X_1 \cup X_2) &= v_i(X_1) + v_i(X_2) - v_i(X_1 \cap X_2) \\
&\geq v_i(X_1) + v_i(X_2) - 2 \cdot L\left(v_i, 2^{k-1}, X_1 \cap X_2\right) \qquad \text{(Since } |X_1 \cap X_2| \leq n = 2^k\text{)} \\
&\geq \frac{1}{2}v_i(M) + L\left(v_i, 2^{k-1}, X_1\right) + \frac{1}{2}v_i(M) + L\left(v_i, 2^{k-1}, X_2\right) - 2 \cdot L\left(v_i, 2^{k-1}, X_1 \cap X_2\right) \\
&> \frac{1}{2}v_i(M) + \frac{1}{2}v_i(M) = v_i(M) > v_i(X_1 \cup X_2)
\end{aligned}$$

which leads to contradiction and concludes the proposition. □

Similarly, we have the same conclusions for $\mathcal{U}_i$. Consider the Kneser graph $\mathcal{H} = \mathcal{K}\left(m, \frac{m}{2}, n\right)$. According to Lemma 1, $\chi(\mathcal{H}) \geq m - 2 \cdot \frac{m}{2} + 2n + 2 = 2n + 2 > 2n$.

As we have proved in proposition 2, for each of $\mathcal{V}_1, \ldots, \mathcal{V}_n, \mathcal{U}_1, \ldots, \mathcal{U}_n$, it does not contain two adjacent vertices of $\mathcal{H}$ and is thus an independent set. Since the number of these sets, $2n$, is less than $\chi(\mathcal{H})$, the union of these $2n$ sets cannot cover all the $\frac{m}{2}$-subsets of M. For this reason, there exists a $\frac{m}{2}$-subset X_0 not belonging to any of $\mathcal{V}_1, \ldots, \mathcal{V}_n, \mathcal{U}_1, \ldots, \mathcal{U}_n$. Therefore, it is not hard to verify $(X_0, M \setminus X_0)$ is a 2-balanced PROP-$\left(\frac{n}{2}, \frac{n}{2}\right)$ allocation. □

Lemma 9. *For any 2-partition (N_1, N_2) such that $|N_1| = \left\lfloor \frac{n}{2} \right\rfloor, |N_2| = \left\lceil \frac{n}{2} \right\rceil$, there always exists a 2-balanced PROP-$(n-1, n)$ allocation which can be computed in polynomial time.*

Proof. For each item $g_j \in M$, we use one decision variable x_j to represent the fraction of item g_j allocated to group N_1. Consider the following linear program:

$$\begin{aligned}
\textbf{max} \quad & \sum_{g_j \in M} v_1(g_j) \cdot x_j - \tfrac{\lfloor n/2 \rfloor}{n} v_1(M) \\
\textbf{subject to} \quad & \sum_{g_j \in M} u_i(g_j) \cdot x_j \geq \tfrac{\lfloor n/2 \rfloor}{n} \cdot u_i(M), && i \in N_1 \\
& \sum_{g_j \in M} v_i(g_j) \cdot x_j \geq \tfrac{\lfloor n/2 \rfloor}{n} \cdot v_i(M), && i \in N_1 \setminus \{1\} \\
& \sum_{g_j \in M} u_i(g_j) \cdot x_j \leq \tfrac{\lfloor n/2 \rfloor}{n} \cdot u_i(M), && i \in N_2 \\
& \sum_{g_j \in M} v_i(g_j) \cdot x_j \leq \tfrac{\lfloor n/2 \rfloor}{n} \cdot v_i(M), && i \in N_2 \\
& 0 \leq x_j \leq 1, && j = 1, \ldots, m.
\end{aligned}$$

Denote the feasible set by Ω. Obviously $(x_1, x_2, \ldots, x_m) = \left(\frac{\lfloor n/2 \rfloor}{n}, \frac{\lfloor n/2 \rfloor}{n}, \ldots, \frac{\lfloor n/2 \rfloor}{n}\right)$ belongs to Ω. Since the objective function's value of $\left(\frac{\lfloor n/2 \rfloor}{n}, \frac{\lfloor n/2 \rfloor}{n}, \ldots, \frac{\lfloor n/2 \rfloor}{n}\right)$ is 0, the optimum of the linear program is nonnegative. Notice that there also exists an optimal solution at a vertex of Ω. Denote it by $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_m^*)$.

Since $\mathbf{x}^*$ is a vertex, according to the definition of vertex, there are at least m constraints that are tight at $\mathbf{x}^*$. Since there are totally $2n - 1 + m$ constraints, at least $m - (2n - 1)$ of the last m constraints are tight. In other words, at least $m - (2n - 1)$ of $x_1^*, \ldots, x_m^*$ are binary (0 or 1). Without loss of generality, we assume $x_1^*, \ldots x_t^* \in (0, 1)$ and $1 \leq t \leq 2n - 1$. Let O_1 and O_2 be $\left\{g_j \in M : x_j^* = 1\right\}$ and $\left\{g_j \in M : x_j^* = 0\right\}$. Then let X_0 and X_1 be defined as

$$X_1 \triangleq \{g_j \in M : j \leq \lceil t/2 \rceil\} \cup O_1, X_2 \triangleq M \setminus X_0.$$

Now we argue that (X_1, X_2) is already 2-balanced PROP-n. For each agent $i \in N_1$,

$$\begin{aligned}
v_i(X_1) &= \sum_{j \leq \lceil t/2 \rceil} v_i(g_j) + \sum_{x_j^* = 1} v_i(g_j) \geq \sum_{j \leq \lceil t/2 \rceil} v_i(g_j) x_j^* + \sum_{x_j^* = 1} v_i(g_j) x_j^* \\
&= \sum_{g_j \in M} v_i(g_j) x_j^* - \sum_{\lceil t/2 \rceil < j \leq t} v_i(g_j) x_j^* \geq \frac{\lfloor n/2 \rfloor}{n} \cdot v_i(M) - L\left(v_i, \lfloor t/2 \rfloor, X_2\right).
\end{aligned}$$

Similarly, we can also have the same conclusion for u_i. For each agent $i \in N_2$,

$$\begin{aligned}
v_i(X_2) &= \sum_{\lceil t/2 \rceil < j \leq t} v_i(g_j) + \sum_{x_j^* = 0} v_i(g_j) \geq \sum_{\lceil t/2 \rceil < j \leq t} v_i(g_j)(1 - x_j^*) + \sum_{j > t} v_i(g_j)(1 - x_j^*) \\
&\geq \sum_{j \in M} v_i(g_j)(1 - x_j^*) - \sum_{j \leq \lceil t/2 \rceil} v_i(g_j) \geq v_i(M) - \sum_{j \in M} v_i(g_j) x_j^* - \sum_{j \leq \lceil t/2 \rceil} v_i(g_j) \\
&\geq v_i(M)\left(1 - \frac{\lfloor n/2 \rfloor}{n}\right) - L(v_i, n, X_1) = \frac{\lceil n/2 \rceil}{n} \cdot v_i(M) - L(v_i, n, X_1).
\end{aligned}$$

Overall, (X_1, X_2) satisfies the definition of 2-balanced PROP-$(n-1, n)$ allocation.

Finally, by Lemma 5, (X_1, X_2) can be computed in polynomial time. □

Remark 2. Notice that we arbitrarily allocate the fractional items to the n agents above. However, we can allocate the fractional items more carefully by utilizing the specific value x_t^* instead of allocating these items arbitrarily. In the following proof, we show that, if $n = 2$, how we allocate the 3 fractional items according to the specific fractional value, and thus achieving PROP-1, corresponding to Theorem 1.

Proof of Theorem 3. Following the proof of Lemma 9, we formulate the problem as a linear program, and we have t variables in $(0, 1)$ for $t \leq 3$.

When $t = 3$, consider the following cases of x_1^*, x_2^*, x_3^*:

– At least two of them are no less than $\frac{1}{2}$, assume $x_1^*, x_2^* \geq \frac{1}{2}$. Consider the allocation $(O_1 \cup \{g_1, g_2\}, O_2 \cup \{g_3\})$. For agent 1, for each $v \in \{u_1, v_1\}$, we have $v(A_1) + v(g_3) \geq \sum_{g_j \in M} v(g_j) x_j^* \geq v(A_2)$. For agent 2, for each $v \in \{u_2, v_2\}$, assume $v(g_1) \geq v(g_2)$, then we have

$$
\begin{aligned}
v(A_2) + v(g_1) &= v(A_2) + (1 - x_1^*) v(g_1) + x_1^* v(g_1) \geq v(A_2) + (1 - x_1) v(g_1) + (1 - x_2^*) v(g_1) \\
&\geq v(A_2) + (1 - x_1^*) v(g_1) + (1 - x_2^*) v(g_2) \geq \sum_{g_j \in M} v(g_j) \left(1 - x_j^*\right) \geq \frac{1}{2} v(M) \geq v(A_1)
\end{aligned}
$$

– At least two of them are no more than $\frac{1}{2}$, assume $x_1^*, x_2^* \leq \frac{1}{2}$. Similar to the above explanation, we can also verify that $(O_1 \cup \{g_3\}, O_2 \cup \{g_1, g_2\})$ is doubly EF-2.

The case for $t \leq 2$ is straightforward. □

Proposition 4. *For an agent i and three given integers $n_1, k_1, k_2 \in \mathbb{Z}^+$, if there exists a set X such that $v_i(A_i) \geq \frac{1}{n_1} v_i(X) - L(v_i, k_1, X)$ and $v_i(X) \geq \frac{n_1}{n} v_i(M) - L(v_i, n_1 \cdot k_2, M \setminus X)$, then $v_i(A_i) \geq \frac{1}{n} v_i(M) - L(v_i, k_1 + k_2, M)$.*

Proof. This proposition can be concluded by the following inequalities,

$$
\begin{aligned}
v_i(A_i) &\geq \frac{1}{n_1} v_i(X) - L(v_i, k_1, X) \geq \frac{1}{n_1} \left(\frac{n_1}{n} v_i(M) - L(v_i, n_1 \cdot k_2, M \setminus X) \right) - L(v_i, k_1, X) \\
&= \frac{1}{n} v_i(M) - \frac{1}{n_1} L(v_i, n_1 \cdot k_2, M \setminus X) - L(v_i, k_1, X) \geq \frac{1}{n} v_i(M) - L(v_i, k_1 + k_2, M).
\end{aligned}
$$

□

Theorem 4. *For any $n = 2^k, k \in \mathbb{Z}_{\geq 0}$, there always exists a doubly PROP-k allocation.*

Proof. We prove this by induction on k. When $k = 0$, this theorem obviously holds. If this theorem holds for any $k \leq k_1$, consider the case of $k = k_1 + 1$. First partition the agents into two groups $N_1 = \{1, \ldots, 2^{k_1}\}$ and $N_2 = \{2^{k_1+1}, \ldots, n\}$. According to Lemma 8, there exists a 2-balanced PROP-$(\frac{n}{2}, \frac{n}{2})$ allocation (X_1, X_2). Next, we consider allocating X_1 to the first group and X_2 to the second group.

By the induction hypothesis, for the agent set N_1 and item set X_1, there exists a doubly PROP-k_1 allocation $\left(A_1, A_2, \ldots, A_{2_1^k}\right)$. Likewise, there also exists a doubly PROP-k_1 allocation $\left(A_{2^{k_1}+1}, \ldots, A_n\right)$ for N_2 and X_2. Respectively apply Proposition 4, we can verify that $v_i(A_i) \geq \frac{1}{n} v_i(M) - L(v_i, k_1 + 1, M)$. Thus, the proof of the induction step is complete. □

Theorem 5. *For any $n \geq 2$, there always exists a doubly PROP-$(2 \lceil \log n \rceil)$ allocation and it can be computed in polynomial time.*

Proof. Similar to the above inductive proof, we first partition the agent set into two groups $N_1 = \{1, \ldots, \lfloor n/2 \rfloor\}$ and $N_2 = \{\lfloor n/2 \rfloor + 1, \ldots, n\}$. According to Lemma 9, there exists a 2-balanced PROP-$(n-1, n)$ allocation (X_1, X_2).

Then, by the same inductive arguments as before, there exist two allocations $\left(A_1, A_2, \ldots, A_{\lfloor n/2 \rfloor}\right)$ and $\left(A_{\lfloor n/2 \rfloor + 1}, \ldots, A_n\right)$, which are respectively PROP-$2 \lceil \log(\lfloor n/2 \rfloor) \rceil$ and PROP-$2 \lceil \log(\lceil n/2 \rceil) \rceil$ for N_1, X_1 and N_2, X_2.

Since $n - 1 \leq 2 \cdot \lfloor n/2 \rfloor$ and $n \leq 2 \cdot \lceil n/2 \rceil$, by applying Proposition 4, we can verify the allocation $(A_1, A_2, \ldots, A_n)$ is doubly PROP-$(2 \lceil \log(\lceil n/2 \rceil) \rceil + 2)$. It is not hard to verify that $\lceil \log(\lceil n/2 \rceil) \rceil = \lceil \log n \rceil - 1$. Therefore, this allocation is also doubly PROP-$(2 \lceil \log n \rceil)$. □

3.4 Binary Valuations

As we have shown in Theorem 5, for general additive valuation, when $n \geq 2$, doubly PROP-$O(\log n)$ allocations always exist. In this section, we further consider another common setting where all the utility functions are binary. We show that a doubly PROP-2 allocation always exists and can be found in polynomial time for any $n \geq 2$ in Theorem 6. The advantage of the binary setting is that, an agent i only needs to focus on the items whose values are regarded as 1 by $v_i(\cdot)$ or $u_i(\cdot)$.

Theorem 6. *When u_i, v_i are both binary for any $i \in N$, there always exists a doubly PROP-2 allocation for any $n \geq 2$ and it can be computed in polynomial time.*

Proof. For each agent $i \in N$, we define the following three item sets: $\mathcal{I}_i^{(1)} \triangleq \{g \in M : v_i(g) = 1 \wedge u_i(g) = 0\}$, $\mathcal{I}_i^{(2)} \triangleq \{g \in M : v_i(g) = u_i(g) = 1\}$, $\mathcal{I}_i^{(3)} \triangleq \{g \in M : u_i(g) = 1 \wedge v_i(g) = 0\}$. Then, we formulate this problem by a linear program. For each agent $i \in N$, we define a vector $\mathbf{x}_i = (x_{i,j})_{j\in[m]}$, where $x_{i,j}$ represents the fraction of item g_j allocated to agent i. Denote $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ by $\mathbf{x}$. Hence $\mathbf{x}$ is a vector with $n \times m$ variables. Consider the polytope $P = \{\mathbf{x} : \mathbf{A}\mathbf{x}^\intercal \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, where $\mathbf{A} \in \mathbb{R}^{(3n+m)\times(n\times m)}$ and $\mathbf{A}\mathbf{x}^\top \leq \mathbf{b}$ is decomposed into two parts:

– For each agent $i \in N$ and $k \in \{1, 2, 3\}$, $\sum_{j\in\mathcal{I}_i^{(k)}} x_{i,j} \geq \left\lfloor 1/n \cdot \left|\mathcal{I}_i^{(k)}\right|\right\rfloor$.
– For each item $g_j \in M$, $\sum_{i\in N} x_{i,j} \leq 1$.

The second part says that a total amount of at most one unit can be allocated for each item j. The first part gives a sufficient condition for the allocation being PROP-2. Specifically, for each agent i, it implies $1 + \sum_{j\in\mathcal{I}_i^{(k)}} x_{i,j} \geq 1/n \cdot \left|\mathcal{I}_i^{(k)}\right|$ for $k = 1, 2, 3$. For $k = 1, 2$, this implies the allocation is PROP-2 with respect to v_i, $2 + \sum_{j\in\mathcal{I}_i^{(1)}} x_{i,j} + \sum_{j\in\mathcal{I}_i^{(2)}} x_{i,j} \geq 1/n \cdot \left(\left|\mathcal{I}_i^{(1)}\right| + \left|\mathcal{I}_i^{(2)}\right|\right)$; for $k = 2, 3$, this implies the allocation is PROP-2 with respect to u_i, $2 + \sum_{j\in\mathcal{I}_i^{(2)}} x_{i,j} + \sum_{j\in\mathcal{I}_i^{(3)}} x_{i,j} \geq 1/n \cdot \left(\left|\mathcal{I}_i^{(2)}\right| + \left|\mathcal{I}_i^{(3)}\right|\right)$.

Notice that $\mathbf{A}$ can also be written as the form $\begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}$, where $\mathbf{A}_1$ and $\mathbf{A}_2$ correspond to the two parts of the constraints. It is easy to verify that $\mathbf{A}_1$ is a matrix containing only 0 and -1 and $\mathbf{A}_2$ is a matrix containing only 0 and 1. Moreover, each column of $\mathbf{A}_1$ and $\mathbf{A}_2$ contains at most one non-zero entry. According to Lemma 3, $\begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}$ is TUM.

Since $\mathbf{x} = (x_{i,j}) = \left(\frac{1}{n}\right)$ is in the polytope, P is nonempty. In addition, since A is TUM and $\mathbf{b}$ is an integer vector, by Lemma 4, all vertices of P are integral. By Lemma 5, we can find a vertex $\mathbf{x}^* \in \{0, 1\}^{n\times m}$ in polynomial time. Then for each agent $i \in N$, allocate the bundle $A_i = \left\{g_j \in M : x^*_{i,j} = 1\right\}$ to her.

Thus, by the definition of the above linear program,

$$v_i(A_i) \geq v_i\left(\mathcal{I}_i^{(1)}\right) + v_i\left(\mathcal{I}_i^{(2)}\right) \geq \frac{1}{n}\left|\mathcal{I}_i^{(k)}\right| - 1 + \frac{1}{n}\left|\mathcal{I}_i^{(k)}\right| - 1 = \frac{v_i(M)}{n} - 2.$$

For the similar reason, we can verify the above inequality for u_i. If there are no less than two items with value 1 outside A_i, this allocation is already PROP-2. Otherwise, if there is at most one item with value 1 outside A_i, then $v_i(M) \leq v_i(A_i) + 1$. It is easy to verify any bundle A_i satisfying this condition is PROP-2. □

4 Allocator's Efficiency

In this section, we consider the problem of maximizing allocator's efficiency subject to EF-c or PORP-c constraint for the agents. Other than general additive utility functions, we also consider the special case of binary utility functions. Note that we no longer consider the special case with identical allocator's utility $u_1 = \cdots = u_n$ since the problem becomes trivial otherwise (all allocations have the same allocator's efficiency).

4.1 Maximizing Allocator's Efficiency for Two Agents

Theorem 7. *The problem of maximizing allocator's efficiency subject to EF-c for two agents is NP-hard to approximate to within factor 2 even when the allocator's utility functions are binary and $c = 1$.*

Proof. We will present a reduction from partition. Given a partition instance $S = \{e_1, \ldots, e_m\}$ such that $\sum_{k=1}^m e_k = 1$, we construct an instance shown in the tables below.

	g_k $(1 \leq k \leq m)$	g_{m+1}	g_{m+2}
v_1	e_k	1	0
v_2	e_k	0	1

Table 4. Agents' Utility Functions

	g_k $(1 \leq k \leq m)$	g_{m+1}	g_{m+2}
u_1	0	0	1
u_2	0	1	0

Table 5. Allocator's Utility Functions

We can observe 2 is an upper bound of allocator's efficiency. If the partition instance is a yes-instance, S can be partitioned into S_1 and S_2 such that $\sum_{e_k \in S_1} e_k = \sum_{e_k \in S_2} e_k = \frac{1}{2}$. The allocation $A_1 = S_1 \cup \{g_{m+2}\}, A_2 = S_2 \cup \{g_{m+1}\}$ satisfies EF-1, and the allocator's efficiency is 2.

If the partition instance is a no-instance, assume the allocator's efficiency is still 2, then the allocation should be $A_1 = S_1' \cup \{g_{m+2}\}, A_2 = S_2' \cup \{g_{m+1}\}$, where $S_1' \cup S_2' = S$. To make this allocation EF-1, we have $\sum_{e_k \in S_1'} \geq \sum_{e_k \in S_2'}$ for agent 1 and $\sum_{e_k \in S_1'} \leq \sum_{e_k \in S_2'}$ for agent 2. Then $\sum_{e_k \in S_1'} e_k = \sum_{e_k \in S_2'} e_k = \frac{1}{2}$, which leads to a contradiction. So the allocator's efficiency is at most 1.

Thus, the inapproximation factor is 2. □

Theorem 8. *The problem of maximizing allocator's efficiency subject to EF-c for two agents has a polynomial time 2-approximation algorithm when the agents' utility functions are arbitrary.*

We first introduce our algorithm. We initialize two empty bundles S_1 and S_2, and sort the items according to agent 1's utility in descending order. Assume the sorted items are $\{g_1, \ldots, g_m\}$, and use $G_i (i \geq 1)$ to denote a group of two items $\{g_{2i-1}, g_{2i}\}$. For each group $G_i (i \geq 1)$, we allocate one item to each bundle. In particular, without loss of generality, we assume $v_2(S_1) \geq v_2(S_2)$ before allocating group G_i. Then, if $v_2(g_{2i-1}) \geq v_2(g_{2i})$, we allocate g_{2i-1} to S_2 and g_{2i} to S_1. Otherwise, we allocate g_{2i-1} to S_1 and g_{2i} to S_2. Notice that, in this algorithm, agent 1's utility function is used exclusively for the ordering of the item, and agent 2's utility function is used exclusively for deciding the allocation of the two items in each group.

After all the items are allocated, we consider the two allocations (S_1, S_2) and (S_2, S_1), and output the allocation with a higher allocator's efficiency.

Proof. First, we show both (S_1, S_2) and (S_2, S_1) satisfy EF-1, and thus satisfy EF-c. For agent 1, by taking $u = v_1$, the same arguments in the proof of Lemma 7 show that the allocation is EF-1 no matter which of S_1 or S_2 she takes.

We prove the allocations are EF-1 to agent 2 by induction. When S_1 and S_2 are empty, both (S_1, S_2) and (S_2, S_1) are trivially EF-1. Assume before allocating group G_i, both allocations satisfy EF-1. Without loss of generality, assume $v_2(S_1) \geq v_2(S_2)$ and $v_2(g_{2i-1}) \geq v_2(g_{2i})$. By our algorithm, after allocating G_i, $S_1' = S_1 \cup \{g_{2i}\}$ and $S_2' = S_2 \cup \{g_{2i-1}\}$. If agent 2 receives S_1', we have

$$v_2(S_1') \geq v_2(S_1) \geq v_2(S_2) = v_2\left(S_2' \setminus \{g_{2i-1}\}\right).$$

If agent 2 receives S_2', since there exists an item $g \in S_1$ such that $v_2(S_2) \geq v_2(S_1 \setminus \{g\})$, we have

$$v_2(S_2') = v_2(S_2) + v_2(g_{2i-1}) \geq v_2\left(S_1 \setminus \{g\}\right) + v_2(g_{2i}) = v_2(S_1' \setminus \{g\}).$$

Hence, both allocations are EF-1.

We next show the allocation with a higher allocator's efficiency is a 2-approximation to the optimal allocator's efficiency. Without loss of generality, assume $\texttt{EFFICIENCY}((S_1, S_2)) \geq \texttt{EFFICIENCY}((S_2, S_1))$. Denote the optimal allocator's efficiency by $\texttt{EFFICIENCY}_{\mathrm{OPT}}$, and we have

$$\begin{aligned} \texttt{EFFICIENCY}_{\mathrm{OPT}} &\leq u_1(M) + u_2(M) = u_1(S_1) + u_2(S_2) + u_1(S_2) + u_2(S_1) \\ &= \texttt{EFFICIENCY}((S_1, S_2)) + \texttt{EFFICIENCY}((S_2, S_1)). \end{aligned}$$

Then, we have $\texttt{EFFICIENCY}((S_1, S_2)) \geq \frac{1}{2} \texttt{EFFICIENCY}_{\mathrm{OPT}}$.

Since the algorithm outputs the allocation (S_1, S_2), Theorem 8 holds. □

4.2 Maximizing Allocator's Efficiency for Constant Number of Agents

Theorem 9. *The problem of maximizing allocator's efficiency subject to EF-c for any fixed $n \geq 3$ is NP-hard to approximate to within any factor that is smaller than $\left\lfloor \frac{1+\sqrt{4n-3}}{2} \right\rfloor$ even when the allocator's utility functions are binary and $c = 1$.*

Proof. We adopt the reduction from partition by [14]. In the origin reduction, the key point is that there exists a super agent, and the social welfare almost depends exclusively on the super agent. In addition, the super agent's utility functions are binary in Bu et al.'s reduction.

In our problem, we maintain the construction in the origin reduction and add an allocator. For the super agent, we set the allocator's utility functions the same as the super agent's utility functions, For other agents, we set the allocator's utility functions to be 0. Hence, the allocator's efficiency is equivalent to the social welfare in the origin reduction, so we get the same inapproximation result. □

Theorem 10. *The problem of maximizing allocator's efficiency subject to EF-c for any fixed $n \geq 3$ can be found in polynomial time when the agents' utility functions are binary.*

Proof. We can adopt the proof of Theorem 7.5 in [4]. In their paper, they used the state of the form $(k, (t_{ij})_{i \neq j}; (b_{ij})_{i \neq j})$ to state whether there exists such an allocation $\mathcal{A} = (A_1, \dots, A_n)$ of $(g_1, \dots, g_k)$ that $v_i(A_i) - v_i(A_j) = t_{ij}$ holds for every two agents $i, j \in N$ and item $g_{b_{ij}}$ is the item which maximizes v_i in agent j's bundle.

The difference from their algorithm is that our state of the form $(k, (t_{ij})_{i \neq j}; (b_{ij})_{i \neq j})$ stores not only the information of the existence, but also the largest value of $\sum_{i \in [n]} u_i(A_i)$ for all satisfying allocations. Besides, we do not need the information b_{ij} as the utility functions are binary. In particular, an allocation is EF-c if and only if $v_i(A_i) - v_i(A_j) \leq t_{ij}$.

We can see that, for the state of the form $(k, (t_{ij})_{i \neq j})$, the values of these parameters can determine the feasibility of the following allocation, and if we keep finding the largest allocator's efficiency among all (partial) allocations stored in this state, we can find the EF-c allocation with the largest allocator's efficiency at the end. □

4.3 Maximizing Allocator's Efficiency for General Number of Agents

For general number of agents, we first consider EF-1, and show a strong inapproximation result even if both the agents' and the allocator's utility functions are binary.

Theorem 11. *For any $\epsilon > 0$, the problem of maximizing allocator's efficiency subject to EF-c is NP-hard to approximate within factor $m^{1-\epsilon}$ or $n^{1/2-\epsilon}$, even if both the agents' and the allocator's utility functions are binary and $c = 1$.*

Proof. We will present a reduction from the maximum independent set problem. For a maximum independent set instance $G = (V, E)$ where $|V| = m$ and $|E| = n$, we construct the following maximizing allocator's efficiency instance with m items, $n + 1$ agents and an allocator. For each vertex $v \in V$, we construct an item g_v. For each edge $e = (u, v) \in E$, we construct a normal agent a_e, whose values to her adjacent items g_u and g_v are 1, and 0 for other items. Moreover, we construct a super agent a_0, whose value to all the items is 0. For the allocator, her value is 1 if an item is allocated to the super agent, and 0 otherwise. We show that the maximum allocator's efficiency is k if and only if the maximum independent set in G is of size k.

If G contains an independent set $\mathcal{I}$ of size k, the maximum allocator's efficiency is at least k by allocating the items that correspond to the vertices in the independent set to the super agent. For normal agents, we allocate at most one adjacent item to her. This allocation is valid since $|E| > |V \setminus \mathcal{I}|$. We now prove the allocation is EF-1. For the super agent a_0, she will envy no one. For an arbitrary normal agent i, she will not envy a_0 because a_0 receives at most one of her adjacent items. She will not envy another normal agent either for the same reason.

If G contains no independent set whose size is larger than k, the maximum allocator's efficiency cannot exceed k. Otherwise, there must exist an edge $e = (u, v)$ that both g_u and g_v are allocated to a_0, and a_e will envy a_0 even if g_u or g_v is removed from a_0's bundle. So the allocation is not EF-1.

Since the maximum independent set problem is known to be NP-hard to approximate to within a factor of $n^{1-\varepsilon}$ and $m = O(n^2)$, Theorem 11 holds. □

We show that a simple variant of round-robin algorithm can achieve m-approximation for EF-c allocations.

Theorem 12. *The problem of maximizing allocator's efficiency subject to EF-c has a m-approximation algorithm when both the agents' and the allocator's utility functions are arbitrary.*

Proof. Let the allocator allocates a single item to a single agent with the highest value $u_i(g_j)$ for $1 \le i \le n, 1 \le j \le m$ to agent i. Then the agents use the round-robin algorithm to allocate the remaining items, where agent i receives an item at the end of each round. The allocation is EF-1 (and is thus EF-c) guaranteed by the round-robin algorithm and is a trivial m-approximation to the optimal allocator's efficiency. □

We note that it is an interesting open question whether there is an $O(n)$-approximation algorithm since the impossibility result of $m^{1-\varepsilon}$ as stated in Theorem 11 occurs when $m < n$. We next turn our attention to a weaker notion PROP-c. We first show that, even if the allocator's utility functions are binary, we can still get the following impossibility result.

Theorem 13. *The problem of maximizing allocator's efficiency subject to PROP-c is NP-hard to approximate within factor* 2 *even if the allocator's utility functions are binary and* $c = 1$*.*

Proof. We will present a reduction from the partition problem. For a partition instance $S = \{e_1, e_2, \ldots, e_m\}$, where $\sum_{k=1}^{m} e_k = x$, we construct an instance as follows. Let $n = 2s$ be an even integer. The instance contains n agents and $s \cdot m + n + 2$ items. We first construct s groups of items called *partition items*, where each group contains m items. Denote the items within k-th group by $g_1^{(k)}, \ldots, g_m^{(k)}$. In addition, we also construct other $n+2$ items called *pool items*. They are denoted by $g_{s+1}, \ldots, g_{s+n+2}$.

item	$g_k^{(1)}$ $(1 \le k \le m)$	$\cdots$	$g_k^{(s)}$ $(1 \le k \le m)$	g_{s+1}	g_{s+2}	$\cdots$	g_{s+n-1}	g_{s+n}	g_{s+n+1}	g_{s+n+2}
v_1	e_k	0	0	0	C	C	C	C	C	C
v_2	e_k	0	0	C	0	C	C	C	C	C
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
v_{2s-1}	0	0	e_k	C	C	C	0	C	C	C
v_{2s}	0	0	e_k	C	C	C	C	0	C	C
u_1	0	0	0	1	0	0	0	0	0	0
u_2	0	0	0	0	1	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
u_{2s-1}	0	0	0	0	0	$\cdots$	1	0	0	0
u_{2s}	0	0	0	0	0	$\cdots$	0	1	0	0

The utility functions of the n agents and the allocator are defined in the above table. For each partition items $g_k^{(i)}$, agent $2i-1$ and $2i$ have value e_k while other agents have value 0. The allocator also has value 0 no matter to whom it is allocated. For each pool item $g_{s+k} (1 \le k \le n)$, agent k has value 0 and other agents have value C, where $C = (\frac{n}{2} - 1) \cdot x$. The allocator has value 1 if g_{s+k} is allocated to agent k and 0 otherwise. For item g_{s+n+1} and g_{s+n+2}, all agents has value C while the allocator has value 0.

For a proportional allocation, each agent must receive a bundle with value at least $\frac{(n-1) \cdot x}{2}$.

We observe the upper bound of the allocator's efficiency is n. If the partition instance is a yes-instance, we allocate $\{g_k^{(i)}\}_{1 \le k \le m}$ to agent $2i-1$ and $2i$ such that each of the two agents receives a value of exactly x. We allocate $g_{s+i} (1 \le i \le n)$ to agent i for each $i \in [n]$, and g_{s+n+1} and g_{s+n+2} to some arbitrary agents. Each agent receives at least $\frac{x}{2}$. This allocation is PROP-1 because, if each agent takes an extra item with value C, she will reach the proportional value. The allocator's efficiency is n for this allocation.

If the partition instance is a no-instance, at least s agents receive values that are less than $\frac{x}{2}$ from the partition items. They need to take a pool item with value C to be PROP-1, where the allocator's value is 0 for it. Then, the allocator's efficiency will be at most $n - s + 2$.

Hence, the inapproximation factor is at most 2. □

If the agents' utility functions are binary but not the allocator's, we can use linear programming to prove the following result.

Theorem 14. *When agents' utility functions are binary, the problem of maximizing allocator's efficiency subject to PROP-c can be solved exactly in polynomial time by linear programming.*

Proof. We first model this problem by a linear program. For each agent $i \in N$ and each item $g_j \in M$, we use one decision variable x_{ij} to represent the fraction of item g_j allocated to agent i. We can get the following linear program.

$$\begin{aligned}
&\mathbf{max} \sum_{i\in[n],j\in[m]} u_i(g_j)x_{ij} \\
&\textbf{subject to} \sum_{j\in[m]} v_i(g_j)x_{ij} \geq \left\lceil \frac{1}{n}\sum_{j\in[m]} v_i(g_j)\right\rceil - c, && \forall i \in [n], && (a)\\
&\sum_{i\in[n]} x_{ij} \leq 1, && \forall j \in [m], && (b)\\
&x_{ij} \geq 0, && \forall i \in [n], j \in [m]. && (c)
\end{aligned}$$

Constraints (a) ensure the corresponding fractional allocation is PROP-c, and Constraints (b) ensure the feasibility of the allocation.

Since the feasible region for this linear programming is bounded by Constraints (b), and setting all x_{ij} as $\frac{1}{n}$ is a feasible solution, there exists an optimal solution to this linear program.

With the integral constraint vector and applying Lemma 4 and Lemma 5, it suffices to show the coefficient matrix $\mathbf{A}$ for this linear programming is totally unimodular (TUM). This follows straightforwardly from Lemma 3. □

5 Conclusion and Future Work

In this paper, we initialize the study of a new fair division model that incorporates the allocator's preference. We focused on the indivisible goods setting and mainly studied two research questions based on the allocator's preference: 1) How to find a doubly fair allocation? 2) What is the complexity of the problem of maximizing allocator's efficiency subject to agents' fairness constraint?

We believe this new model is worth more future studies. For example, could we extend our results to the setting with more general valuation functions, e.g., submodular valuations? It is also an interesting (and challenging) problem to study what is the minimum number c where a doubly EF-c/PROP-c allocation is guaranteed to exist. Indeed, we do not know any lower bound to c. In particular, we do not know if a doubly EF-1, or even doubly PROP-1, allocation exists even for binary utility functions. We have searched for a non-existence counterexample with the aid of computer programs, and a non-existence counterexample seems hard to find.

On the other hand, our current techniques about Kneser graph and linear programming seem to have their limitations for further reducing the upper bound of c. Our current technique with Kneser graph can only analyze a bi-partition of the items with an equal size $m/2$ (this is crucial for Proposition 1 and Proposition 2). In addition, the value of the bundle must be exactly *half* of the total value up to the addition of c items. This is why we need n to be an integer power of 2 in Theorem 4. Moreover, the nature of the analysis based on Kneser graph makes the existence proof non-constructive. Our linear programming technique, on the other hand, provides a weaker bound on c. It seems to us that a Kneser graph captures more structural insights about our problem than a linear program. Nevertheless, linear programming-based techniques provide a constructive existence proof.

It is fascinating to see how these techniques can be further exploited and if the above-mentioned limitations can be bypassed. Unearthing new techniques for closing the gap between the upper bound and the lower bound of c may also be necessary.

5.1 Fair Division with Multiple Sets of Valuations

In our double fairness setting, we aim to find an allocation $(A_1, \ldots, A_n)$ that is fair with respect to *two* valuation profiles $(u_1, \ldots, u_n)$ and $(v_1, \ldots, v_n)$, one for the agents and one for the allocator. A natural generalization of this is to consider allocations that are fair with respect to t valuation profiles for general t. The problem of fair division with more than two sets of valuations is also well-motivated in many applications.

For example, there may be more than one "allocator" in many scenarios. Taking the example of educational resource allocation in Sect. 1, the government may consist of multiple parties, and it is desirable to find an allocation that is fair for all parties. For another example, an agent's valuation of the items may be multi-dimensional. When allocating employees to the departments of an organization, fairness is evaluated by multiple factors including employees' salaries, skill sets, diversity, etc. When dealing with multiple sets of valuations, different fairness criteria can be considered.

As a natural generalization of the setting in this paper, we can consider allocations that are EF-c or PROP-c for *all* valuation profiles. This coincides with the setting of group fairness with group sizes satisfying $n_1 = n_2 = \cdots = n_k = t$ (see the last paragraph of Sect. 1.2 for further discussions). In contrast to our results in Theorem 1 and Theorem 2, even when there are only two agents and the valuations of the agents in each of the three profiles $(u_1, u_2), (v_1, v_2), (w_1, w_2)$ are identical (i.e., $u_1 = u_2$, $v_1 = v_2$, and $w_1 = w_2$) and binary, a triply EF-1 allocation may fail to exist. In the example in Table 6, it is easy to see that Items 1 and 2 must not be in the same bundle based on u_1 and u_2, Items 1 and 3 must not be in the same bundle based on v_1 and v_2, and Items 2 and 3 must not be in the same bundle based on w_1 and w_2. Clearly, no allocation satisfies these. It is then natural to ask for which values of c there is always an allocation that is EF-c for all valuation profiles.

	Item 1	Item 2	Item 3
Values based on $u_1 = u_2$	1	1	0
Values based on $v_1 = v_2$	1	0	1
Values based on $w_1 = w_2$	0	1	1

Table 6. An example where a triply EF-1 allocation fails to exist.

Another compelling criterion is to make the allocation fair with respect to ℓ out of k valuation profiles. Using the well-studied criterion EF-1 as an example, considering k valuation profiles

$$\left\{(u_1^{(1)}, \ldots, u_n^{(1)}), (u_1^{(2)}, \ldots, u_n^{(2)}), \ldots, (u_1^{(k)}, \ldots, u_n^{(k)})\right\}$$

and given a parameter $\ell \leq k$, our goal is to find an allocation $(A_1, \ldots, A_n)$ such that, for each agent i, there exists ℓ valuation functions from $\{u_i^{(1)}, \ldots, u_i^{(k)}\}$ such that the allocation satisfies EF-1 with respect to these ℓ valuation functions. It is interesting to find out for which values of k and ℓ this is possible.

Acknowledgments

The research of Biaoshuai Tao was supported by the National Natural Science Foundation of China (No. 62102252). The research of Shengxin Liu was partially supported by the National Natural Science Foundation of China (No. 62102117), by the Shenzhen Science and Technology Program (No. RCBS20210609103900003), and by the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011188), and by CCF-Huawei Populus Grove Fund (No. CCF-HuaweiLK2022005).

References

1. Alon, N.: Splitting necklaces. Advances in Mathematics **63**(3), 247–253 (1987)
2. Amanatidis, G., Aziz, H., Birmpas, G., Filos-Ratsikas, A., Li, B., Moulin, H., Voudouris, A.A., Wu, X.: Fair division of indivisible goods: Recent progress and open questions. Artificial Intelligence (2023), forthcoming
3. Aziz, H.: Developments in multi-agent fair allocation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). pp. 13563–13568 (2020)
4. Aziz, H., Huang, X., Mattei, N., Segal-Halevi, E.: Computing welfare-maximizing fair allocations of indivisible goods. European Journal of Operational Research **307**(2), 773–784 (2023)
5. Aziz, H., Mackenzie, S.: A discrete and bounded envy-free cake cutting protocol for any number of agents. In: Proceedings of the Annual IEEE Symposium on Foundations of Computer Science (FOCS). pp. 416–427 (2016)

6. Aziz, H., Mackenzie, S.: A discrete and bounded envy-free cake cutting protocol for four agents. In: Proceedings of the Annual ACM Symposium on Theory of Computing (STOC). pp. 454–464 (2016)
7. Aziz, H., Moulin, H., Sandomirskiy, F.: A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation. Operations Research Letters **48**(5), 573–578 (2020)
8. Bárány, I.: A short proof of Kneser's conjecture. Journal of Combinatorial Theory, Series A **25**(3), 325–326 (1978)
9. Barman, S., Ghalme, G., Jain, S., Kulkarni, P., Narang, S.: Fair division of indivisible goods among strategic agents. In: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). p. 1811–1813 (2019)
10. Barman, S., Krishnamurthy, S.K.: On the proximity of markets with integral equilibria. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). pp. 1748–1755 (2019)
11. Bei, X., Chen, N., Hua, X., Tao, B., Yang, E.: Optimal proportional cake cutting with connected pieces. In: Proceedings of AAAI Conference on Artificial Intelligence (AAAI). pp. 1263–1269 (2012)
12. Brams, S.J., Feldman, M., Lai, J.K., Morgenstern, J., Procaccia, A.D.: On maxsum fair cake divisions. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). pp. 1285–1291 (2012)
13. Brams, S.J., Taylor, A.D.: An envy-free cake division protocol. The American Mathematical Monthly **102**(1), 9–18 (1995)
14. Bu, X., Li, Z., Liu, S., Song, J., Tao, B.: On the complexity of maximizing social welfare within fair allocations of indivisible goods. arXiv preprint arXiv:2205.14296 (2022)
15. Budish, E.: The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. Journal of Political Economy **119**(6), 1061–1103 (2011)
16. Budish, E., Cachon, G.P., Kessler, J.B., Othman, A.: Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. Operations Research **65**(2), 314–336 (2017)
17. Caragiannis, I., Kurokawa, D., Moulin, H., Procaccia, A.D., Shah, N., Wang, J.: The unreasonable fairness of maximum Nash welfare. ACM Transactions on Economics and Computation **7**(3), 1–32 (2019)
18. Cohler, Y.J., Lai, J.K., Parkes, D.C., Procaccia, A.D.: Optimal envy-free cake cutting. In: Proceedings of AAAI Conference on Artificial Intelligence (AAAI). pp. 626–631 (2011)
19. Conitzer, V., Freeman, R., Shah, N.: Fair public decision making. In: Proceedings of the ACM Conference on Economics and Computation (EC). pp. 629–646 (2017)
20. Even, S., Paz, A.: A note on cake cutting. Discrete Applied Mathematics **7**(3), 285–296 (1984)
21. Foley, D.K.: Resource allocation and the public sector. Yale Economics Essays **7**(1), 45–98 (1967)
22. Freeman, R., Micha, E., Shah, N.: Two-sided matching meets fair division. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI). pp. 203–209 (2021)
23. Goldman, J., Procaccia, A.D.: Spliddit: Unleashing fair division algorithms. ACM SIGecom Exchanges **13**(2), 41–46 (2015)
24. Gollapudi, S., Kollias, K., Plaut, B.: Almost envy-free repeated matching in two-sided markets. In: Proceedings of the International Conference on Web and Internet Economics (WINE). pp. 3–16 (2020)
25. Greene, J.E.: A new short proof of Kneser's conjecture. The American Mathematical Monthly **109**(10), 918–920 (2002)
26. Güler, O., den Hertog, D., Roos, C., Terlaky, T., Tsuchiya, T.: Degeneracy in interior point methods for linear programming: A survey. Annals of Operations Research **46**(1), 107–138 (1993)
27. Hoffman, A.J., Kruskal, J.B.: Integral boundary points of convex polyhedra. In: 50 Years of Integer Programming 1958-2008, pp. 49–76 (2010)
28. Igarashi, A., Kawase, Y., Suksompong, W., Sumita, H.: Fair division with two-sided preferences. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI) (2023), forthcoming
29. Jafari, A., Moghaddamzadeh, M.J.: On the chromatic number of generalized kneser graphs and hadamard matrices. Discrete Mathematics **343**(2), 111682 (2020)
30. Kyropoulou, M., Suksompong, W., Voudouris, A.A.: Almost envy-freeness in group resource allocation. Theoretical Computer Science **841**, 110–123 (2020)
31. Lipton, R., Markakis, E., Mossel, E., Saberi, A.: On approximately fair allocations of indivisible goods. In: Proceedings of the ACM Conference on Electronic Commerce (EC). pp. 125–131 (2004)
32. Liu, S., Lu, X., Suzuki, M., Walsh, T.: Mixed fair division: A survey. arXiv preprint arXiv:2306.09564 (2023)
33. Lovász, L.: Kneser's conjecture, chromatic number, and homotopy. Journal of Combinatorial Theory, Series A **25**(3), 319–324 (1978)
34. Manurangsi, P., Suksompong, W.: Asymptotic existence of fair divisions for groups. Mathematical Social Sciences **89**, 100–108 (2017)
35. Manurangsi, P., Suksompong, W.: Almost envy-freeness for groups: Improved bounds via discrepancy theory. Theoretical Computer Science **930**, 179–195 (2022)
36. Matoušek, J.: A combinatorial proof of Kneser's conjecture. Combinatorica **24**(1), 163–170 (2004)

37. McGlaughlin, P., Garg, J.: Improving Nash social welfare approximations. Journal of Artificial Intelligence Research **68**, 225–245 (2020)
38. Moulin, H.: Fair division in the internet age. Annual Review of Economics **11**(1), 407–441 (2019)
39. Patro, G.K., Biswas, A., Ganguly, N., Gummadi, K.P., Chakraborty, A.: Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In: Proceedings of the Web conference (WWW). pp. 1194–1204 (2020)
40. Procaccia, A.D.: Cake cutting: Not just child's play. Communications of the ACM **56**(7), 78–87 (2013)
41. Robertson, J., Webb, W.: Cake-Cutting Algorithm: Be Fair If You Can. A K Peters/CRC Press (1998)
42. Segal-Halevi, E., Nitzan, S.: Fair cake-cutting among families. Social Choice and Welfare **53**(4), 709–740 (2019)
43. Segal-Halevi, E., Suksompong, W.: Democratic fair allocation of indivisible goods. Artificial Intelligence **277**, 103167 (2019)
44. Steinhaus, H.: The problem of fair division. Econometrica **16**(1), 101–104 (1948)
45. Steinhaus, H.: Sur la division pragmatique. Econometrica **17**, 315–319 (1949)
46. Suksompong, W.: Approximate maximin shares for groups of agents. Mathematical Social Sciences **92**, 40–47 (2018)
47. Suksompong, W.: Constraints in fair division. ACM SIGecom Exchanges **19**(2), 46–61 (2021)
48. Varian, H.R.: Equity, envy, and efficiency. Journal of Economic Theory **9**(1), 63–91 (1974)

A Divisible Resources with Allocator's Preference

In this section, we discuss the two research questions mentioned before Sect. 1.1 for divisible resources. The two fairness notions envy-freeness and proportionality discussed in this paper can be satisfied exactly in the setting with divisible resources. Thus, we focus on exact envy-freeness and proportionality here. There are multiple different models for divisible resources.

A.1 Divisible Homogeneous Items

The simplest setting is the same setting as it is in Sect. 2 except that we now allow fractional allocations, i.e., each $g_j \in M$ can now be split among the agents. Each item g_j is assumed to be homogeneous: each agent i's value on an α-fraction of g_j is given by $\alpha \cdot v_i(\{g_j\})$.

For the first research question, there exists a trivial doubly envy-free and doubly proportional allocation: just allocate each item evenly to the agents such that each agent gets a $1/n$ fraction of each item.

For the second research question, the problem of maximizing allocator's efficiency subject to the envy-free/proportional constraint can be formulated by a linear program. Let x_{ij} be the fraction of item j allocated to agent i. It is straightforward to see that the envy-free constraints and the proportional constraints are linear in x_{ij}'s, and the allocator's efficiency is also a linear expression of x_{ij}'s. This gives us a polynomial time algorithm to solve this constrained optimization problem exactly.

A.2 Cake Cutting

Another well-studied model for divisible resources is the *cake-cutting* model. In the cake-cutting model, a single piece of heterogeneous resource, modeled by the interval $[0, 1]$, is allocated to n agents. Each agent i has a *value density function* $f_i : [0, 1] \to \mathbb{R}_{\geq 0}$, and her value of a subset $S \subseteq [0, 1]$ is given by the Riemann integral

$$\int_S f_i(x)dx.$$

The fairness notions envy-freeness and proportionality can then be defined accordingly.

In computer scientists' perspective, we then face the problem of succinct representation of each f_i. There are two different approaches in the past literature.

Piecewise-constant value density functions In the first approach, each f_i is assumed to be piecewise-constant, where the interval $[0, 1]$ can be partitioned into many subintervals where f_i is a constant on each of them (see, e.g., [18,11,12]). Piecewise-constant functions can be succinctly represented and can approximate real functions with arbitrarily good precision.

This model then reduces to the previous model: we can find all the points of discontinuity of $f_1, \ldots, f_n$; this will partition $[0, 1]$ into many subintervals where each f_i is a constant on each of them, and each of these subintervals can be viewed as an "item" in the previous model.

Therefore, all results in the previous model apply here. We can find a doubly envy-free (and thus doubly proportional) allocation in polynomial time, and we can solve the problem of maximizing the allocator's efficiency subject to agents' envy-free/proportional constraints in polynomial time by linear programming.

General value density functions If no assumption is made on the value density functions, the existence of a doubly envy-free allocation still holds. [1] showed that for m agents and any positive number n, there exists an allocation $(A_1, \ldots, A_n)$ such that each A_i has value exactly $\frac{1}{n}$ of the value of $[0, 1]$ based on each agent's value density function. By taking $m = 2n$, this implies the existence of a doubly envy-free allocation.

The second problem of maximizing the allocator's efficiency is related to the computational complexity, so we need to define a model to access the value density functions. A commonly used one is the *Robertson-Webb query model* [41]. However, finding an envy-free allocation under this model is already challenging and solved only recently, with an exponential time complexity [5,6].