Theoretical Computer Science 981 (2024) 114246

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Repeatedly matching items to agents fairly and efficiently ☆

Ioannis Caragiannis [a,1], Shivika Narang [b,*,2]

[a] Aarhus University, Denmark
[b] Indian Institute of Science, India

ARTICLE INFO

Communicated by M. Mavronicolas

Keywords:
Resource allocation
Fair division
Envy-freeness
Matchings

ABSTRACT

We consider a novel setting where a set of items is matched to the same set of agents repeatedly over multiple rounds. Each agent gets exactly one item per round, which brings interesting challenges to finding efficient and/or fair *repeated matchings*. A particular feature of our model is that the value of an agent for an item in some round depends on the number of rounds in which the item has been used by the agent in the past. We present a set of positive and negative results about the efficiency and fairness of repeated matchings. For example, when items are goods, an adaptation of the well-studied fairness notion of envy-freeness up to one good (EF1) can be satisfied under certain conditions. Furthermore, it is intractable to achieve fairness and (approximate) efficiency simultaneously, even though they are achievable separately. For mixed items, which can be goods for some agents and chores for others, we propose and study a new notion of fairness that we call *swap envy-freeness* (swapEF).

1. Introduction

The problem of fairly dividing indivisible items among agents has received enormous attention by the EconCS research community in the recent years. The standard setting involves a set of items and agents who have values for them. The objective is to compute an allocation which gives each item to a single agent so that some notion of fairness is satisfied. A diverse set of fairness objectives has been explored in the past; some of the most well known of these are envy-freeness and its relaxations. Prior work has typically explored various settings where agents' allocations do not change with time. Typically, the number of items allocated to an agent is not explicitly restricted, with the exception of some recent work [20,24,11,19]. However, sometimes in practice, the same set of items must be allocated to the same set of agents repeatedly. More crucially, another feature that distinguishes such scenarios from the standard setting is that the value of an agent for an item changes over time and typically depends on how many times the agent has received the item in the past. This can make solutions that were fair when the agents were allocated the items once, no longer fair.

To give an example, consider different research labs that all need access to several expensive research facilities in a university. How should the access of the labs to the facilities be fairly coordinated/scheduled throughout the year? This is a fair division problem

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

* Corresponding author.

E-mail addresses: iannis@cs.au.dk (I. Caragiannis), shivika@iisc.ac.in (S. Narang).

URLs: https://cs.au.dk/~iannis/index.html (I. Caragiannis), https://sites.google.com/view/shivikanarang/home (S. Narang).

[1] Ioannis Caragiannis is partially supported by the Independent Research Fund Denmark (DFF) under grant 2032-00185B.

[2] Shivika Narang is supported by a Tata Consultancy Services Research Fellowship.

https://doi.org/10.1016/j.tcs.2023.114246
Received 17 March 2023; Received in revised form 11 September 2023; Accepted 6 October 2023

Available online 12 October 2023

with the labs and the facilities playing the role of the agents and the items, respectively. To be fair among labs and efficient overall, such a scheduling should take into account the values the labs have for facilities, which typically change over time. For instance, during the first few weeks of access to a facility, the researchers in a lab may need time to learn how to operate it. During that time, the value the lab gets by accessing a facility can be very low, even negative. As the researchers gain more experience, their research output increases, and so does the lab's value for the facility. Once the researchers have run their intended experiments, the lab's value for the facility decreases again until the next experiment.

To capture such situations, we introduce a new model of *repeated matchings* with n agents who must be matched with exactly one of n items[3] in each of T rounds, repeatedly. An important novelty of our model is that valuations are *history-dependent*: the value an agent has for an item in a round depends on how many times the agent has used the item in previous rounds. Such valuations reveal many interesting hurdles to achieving efficiency and fairness. We use *social welfare* (the total value of the agents from the items they get in all rounds) to assess the efficiency of repeated matchings. We also use relaxations of *envy-freeness* as fairness concepts. We adapt the well-known *envy-freeness up to one item* (EF1) and use it when all valuations are non-negative (i.e., when items are *goods*). A repeated matching is EF1 if the value of every agent i for her bundle is at least as high as her value for the bundle of any other agent j after removing the last *copy* of an item from j's bundle. We observe that EF1 is not suitable when valuations can be positive or negative (i.e., when items are *mixed*), and introduce the notion of *swap envy-freeness* to assess fairness of repeated matchings for mixed items.

1.1. Our contribution

More specifically, our technical contribution is as follows. We prove that the problem of computing a repeated matching with maximum social welfare is NP-hard, even when $T = 3$. Our hardness reduction defines instances with non-monotone valuations. The problem becomes solvable in polynomial time when the valuations are monotone. This is when the value an agent has for an item can only decrease or increase, but not both, in terms of the number of rounds the agent had the item in the past. For the case of monotone non-increasing valuations, earlier work on b-matchings can be leveraged to find the optimal solution. When the valuations are monotone non-decreasing, we find a neat reduction to the case of time-constant valuations which can be solved efficiently.

We also consider fair repeated matchings, using EF1 as fairness concept. We find that under identical valuations, EF1 repeated matchings always exist and can be found in polynomial time. Furthermore, we show that any instance with general valuations and $T \bmod n \in \{0, 1, 2, n-1\}$ (i.e., including all instances with at most four agents/items) has an EF1 repeated matching, which can be computed efficiently. We establish that, unfortunately, EF1 is not compatible with social welfare maximization and even approximating the maximum social welfare over EF1 repeated matchings is NP-hard. This holds even for settings where EF1 solutions and social welfare maximizing solutions can be found in polynomial time separately.

Moreover, at a conceptual level, we propose and study a new fairness notion called swap envy-freeness (swapEF). Here, we find that under identical valuations, swapEF repeated matchings can be found using the same algorithm as used for EF1. Furthermore, we show that swapEF repeated matchings always exist and can be computed efficiently on instances with $T \bmod n \in \{0, 1, 2, n-2, n-1\}$ (i.e., including all instances with at most five agents/items). Our hardness results are proved on instances with goods. Our positive results besides those for EF1, apply to instances with mixed items.

1.2. Related work

In fair division with indivisible items (see Amanatidis et al. [2] for a recent survey of the area), EF1 has been established as a key fairness concept. It was defined by Budish [12] (and, implicitly, a few years earlier by Lipton et al. [29]). In contrast to envy-freeness which is usually impossible to achieve, EF1 is always achievable in the standard setting and is also compatible with notions of economic efficiency [14,6]. These papers assume that items are goods, i.e., agents have non-negative valuations for them. Non-positive valuations, i.e., indivisible chores, have also received attention. More importantly, a series of recent papers consider mixed items that can be goods for some agents and chores for others [3,10,9].

The main assumption in the standard setting is that each item is given to exactly one agent with no explicit cap on the number of items one agent can get. Ferraioli et al. [19] consider an allocation problem where all agents must get the same number of items. Biswas and Barman [11] consider an extension where the items are partitioned into categories and there are cardinality constraints on how many items an agent can be allocated from each category. They show how to compute an EF1 allocation by extending the envy-cycle elimination algorithm of Lipton et al. [29]. Even though cardinality constraints can restrict allocations to repeated matchings, our history-dependent valuations cannot be expressed by their model. Another extension is considered by Gafni et al. [22] where each item may have multiple copies. They study relaxations of envy-freeness with mixed items, in a model where each agent can get at most one item copy. Some work has also been done on envy-freeness and its relaxations, in the roommate matching model where agents get value from their rooms as well as roommates [15,28,27]. Here the size of the bundle an agent gets (their room and roommates) is determined by the capacity of the room.

The concept of repeated matching has been considered before, actually using EF1 as fairness concept. However, history-dependent valuations have not. Hosseini et al. [26] look at a dynamic one-sided repeated matching model with ordinal preferences that change over time. They study strategyproofness and give a mechanism that is EF1. As the model of preferences studied is entirely different,

[3] This assumption is without loss of generality since, whenever there are more agents or items, we can add dummy items or agents with zero values, respectively.

their results are not applicable to our model. Gollapudi et al. [24] study a two-sided repeated matching setting where the agent values may change in each round, but do not take into account how often the two agents have been matched in the past. In addition, due to the two-sided nature of their setting, their results are not applicable to our case.

Finally, relaxations of envy-freeness have been considered extensively in the literature. For mixed items in particular, Aleksandrov [1] summarizes the several variations of EF1 that have been proposed in the literature and proposes new ones. Others include EFL, envy-freeness up to one less preferred good [4], EFR, envy-freeness up to a random good [18] and iEF, interim envy-freeness [13]. The setting of identical valuations has also been specifically explored, both for the existence of almost envy-free solutions [30,16] and other fairness objectives [7,8]. To the best of our knowledge, swap envy-freeness appears to be novel.

1.3. Roadmap

The rest of the paper is structured as follows. We begin with setting up the notation and relevant definitions in Section 2. Section 3 focuses on maximizing social welfare. Here, we give our hardness result for maximizing social welfare in general and polynomial-time algorithms for monotone valuations. In Section 4, we explore settings under which we can satisfy EF1 and algorithms that find EF1 solutions. In Section 5, we find that even in settings where EF1 repeated matchings can be found in polynomial time, maximizing social welfare over the space of EF1 repeated matchings is intractable. We devote Section 6 to the study of swap envy-freeness. We conclude with a discussion on open problems in Section 7.

2. Notation and preliminaries

Our setting involves a set $\mathcal{A}$ of n agents and a set $\mathcal{G}$ of n items. We use the term *matching* to refer to an allocation of the items to the agents, so that each agent gets exactly one item and each item is given to exactly one agent. We particularly focus on *repeated matchings*, where the items are matched to the agents in multiple rounds. More formally, we consider instances of the form $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$, where T denotes the number of rounds and, for each agent $i \in \mathcal{A}$, v_i is a function from $\mathcal{G} \times [T]$ to $\mathbb{R}$, where $v_i(g,t)$ denotes the *valuation* of agent i for item g when it is matched to the item for the t^{th} time. A repeated matching $A = (A^1, \dots, A^T)$ is simply a collection of matchings, with one matching A^t per each round $t \in [T]$. Furthermore, we denote by A_i the multiset (or *bundle*) which contains copies of the items to which agent $i \in \mathcal{A}$ is matched in the T rounds.

Hence, defining the bundles A_i for $i \in \mathcal{A}$ given the repeated matching A is trivial. The opposite task is also straightforward. Let $N(B,g)$ be the multiplicity of item g in bundle B. Given bundles of items A_i for $i \in \mathcal{A}$ with $|A_i| = T$ (i.e., each agent gets T copies of items) and $\sum_{i\in\mathcal{A}} N(A_i,g) = T$ (i.e., T copies of each item g are allocated), a consistent repeated matching[4] for instance I is obtained as follows. We construct the bipartite multigraph $G = (\mathcal{A}, \mathcal{G}, E)$ so that the set of edges E consists of (a copy of) edge (i,g) for every (copy of) item g such that $g \in A_i$. The graph G is T-regular and, thus, by Hall's matching theorem (see [31]), can be edge-colored with T colors, e.g., using the polynomial time algorithm of Cole et al. [17]. The edges of the same color correspond to a matching; thus, the edge-coloring can be thought of as decomposing the edges of G into T matchings M_1, ..., M_T. These matchings correspond to a repeated matching by interpreting the edge (i,g) in matching M_t as the assignment of item g to agent i in the t^{th} round. In this manner, given the number of times each agent must be matched to each item, we can generate a corresponding repeated matching. We call this procedure GenerateFromFreq() and shall use it in our algorithms.

With a slight abuse of notation, we use $v_i(B)$ to denote the *value* agent $i \in \mathcal{A}$ has when she gets the bundle B, i.e.,

$$v_i(B) = \sum_{g\in\mathcal{G}} \sum_{t=1}^{N(B,g)} v_i(g,t).$$

Hence, for a repeated matching A, $v_i(A_i)$ is the total value from each item copy agent i receives in all rounds. The *social welfare* of A is simply the sum of the agents' values for their bundle, i.e., $SW(A) = \sum_{i\in\mathcal{A}} v_i(A_i)$.

We shall look at specific types of valuations under which we will try to find efficient and/or fair repeated matchings. A well-motivated setting is that of *identical* valuations where $v_1 = v_2 = \cdots = v_n$. This assumption proves particularly useful in finding fair solutions. Another important class of valuation functions is that of *monotone* valuations.

Definition 1 *(monotone valuations).* The valuation function v_i is monotone non-increasing (respectively, monotone non-decreasing) if for every item $g \in \mathcal{G}$, and $t \in [T-1]$, we have that $v_i(g,t) \ge v_i(g,t+1)$ (respectively, $v_i(g,t) \le v_i(g,t+1)$).

These two classes of valuation functions intersect in the class of *constant* valuations.

Definition 2 *(constant valuations).* Valuation function v_i is said to be constant if for every item $g \in \mathcal{G}$, we have that $v_i(g,1) = v_i(g,2) = \cdots = v_i(g,T) = v_i(g)$.

We extend to repeated matchings the well-known fairness notion of *envy-freeness of up to one item* (EF1) as follows.

[4] We remark that this repeated matching is not unique. However, this does not affect the values of each agent for her bundle and the bundle of any other agent, which are the same in all different consistent repeated matchings.

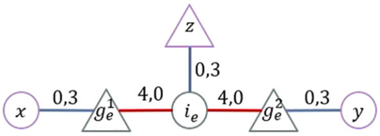

Fig. 1. Agents and items in the repeated matching instance for every edge $e = (x, y, z)$ in the 3DM instance in the proof of Theorem 1. The two labels on each edge denote the valuation for the two item copies.

Definition 3 *(*EF1*)*. A repeated matching A is EF1 if for every pair of agents $i, j \in \mathcal{A}$, either there exists an item $g \in A_j$ such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$, or there exists an item $g \in A_i$ such that $v_i(A_i \setminus \{g\}) \geq v_i(A_j)$.

We remark that the operation $A_j \setminus \{g\}$ removes one copy of item g from the bundle A_j if g belongs to A_j and leaves A_j intact otherwise.

We refer to the items as *goods* on instances where all valuations are non-negative, i.e., when $v_i(g, t) \geq 0$ for every $i \in \mathcal{A}$, $g \in \mathcal{G}$, and $t \in [T]$. When there are no restrictions on the valuations, we refer to the items as *mixed*. The two cases in the definition of EF1 above (Definition 3) handle the cases of items that are goods, chores, or mixed.

3. Maximizing social welfare

We begin our technical exposition by studying the complexity of the problem of computing a repeated matching of maximum social welfare. Notice that if $T = 1$, this task can be easily done by computing a maximum-weight perfect matching in the complete bipartite graph $G = (\mathcal{A}, \mathcal{G}, \mathcal{A} \times \mathcal{G})$, in which edge (i, g) has weight $v_i(g, 1)$. For $T > 1$, an approach that seems natural computes gradually a maximum-weight perfect matching for each round, taking into account the matching decisions in previous rounds.

For example, consider the instance with three agents and two rounds (i.e., $n = 3$, $\mathcal{A} = \{1, 2, 3\}$, $\mathcal{G} = \{g_1, g_2, g_3\}$, and $T = 2$). The agent valuations are as follows: $v_1(g_2, 1) = v_1(g_3, 1) = 1 - \epsilon$ (for small but strictly positive ϵ), $v_2(g_2, 1) = v_3(g_3, 1) = 1$, while all other valuations are 0. A maximum-weight perfect matching on the complete bipartite graph $G = (\mathcal{A}, \mathcal{G}, \mathcal{A} \times \mathcal{G})$ with weight $v_i(g, 1)$ on edge (i, g) assigns item g_i to agent i in the first round; this gives value 1 to agents 2 and 3. Then, the natural way to compute the matching of the second round is to compute a maximum-weight perfect matching in the complete bipartite graph $G = (\mathcal{A}, \mathcal{G}, \mathcal{A} \times \mathcal{G})$ with weight $v_i(g_i, 2)$ to edge (i, g_i) (because agent i already uses item i in the first round) and weight $v_i(g, 1)$ to edge (i, g) for $g \neq g_i$. In this way, the matching of the second round will give value of $1 - \epsilon$ to agent 1 only, by matching her to either item g_2 or item g_3. Thus, the social welfare is $3 - \epsilon$. In contrast, consider the repeated matching in which the first-round matching assigns item g_2 to agent 1, item g_1 to agent 2, and item g_3 to agent 3, and the second-round matching assigns item g_3 to agent 1, item g_2 to agent 2, and item g_1 to agent 3. Agent 1 gets value $1 - \epsilon$ in both rounds, agent 2 gets value 1 in the second round, and agent 3 gets value 1 in the first round. Hence, the social welfare is now $4 - 2\epsilon$.

This example demonstrates that computing a repeated matching of maximum social welfare can be a challenging task. Actually, as our first result indicates, the problem is computationally hard.

Theorem 1. *Given a repeated matching instance, computing a repeated matching of maximum social welfare is NP-hard.*

Proof. We present a polynomial-time reduction from three dimensional matching (3DM). Given a tripartite hypergraph $G = (X, Y, Z, E)$ s.t. $|X| = |Y| = |Z| = k$ and $|E| = p$, we must decide where there exists a subset of size k of edges in E such that each node is incident on exactly one edge. This is a well-known NP-hard problem [23]. Such a subset is a three-dimensional perfect matching. When such a subset exists, we shall say that a 3DM exists. Given such a graph, we shall construct a repeated matching instance such that a 3DM exists if and only if there is a repeated matching of social welfare at least $8p + k$.

We construct an instance $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i \in \mathcal{A}}, T \rangle$ with $T = 2$ and $n = \max(2k + p, k + 2p)$ agents/items. The set of agents $\mathcal{A}$ has a *node agent* x for every $x \in X$ and one *node agent* y for every $y \in Y$. There is also an *edge agent* i_e for every $e = (x, y, z) \in E$. The set of items $\mathcal{G}$ has a *node item* for every node $z \in Z$, and two *edge items* g_e^1 and g_e^2 for every $e \in E$. The remaining items or agents are dummy agents and items, which never give any agent any value or have any value of their own. Let $e = (x, y, z) \in E$. The valuations are as follows:

- Node agent x has value 3 for the second copy of edge item g_e^1.
- Node agent y has value 3 for the second copy of edge item g_e^2.
- Edge agent i_e has value 3 for the second copy of node item z.
- Edge agent i_e has value 4 for the first copy of g_e^1 and g_e^2.
- All other valuations are 0 (including the valuation of any agent for a dummy item or of any dummy agent for any item).

This is illustrated in Fig. 1. Thus, for node agents x or y to get any value, they must be matched to an edge item for both rounds. Analogously for z to generate any value, it must be matched to an edge agent for both rounds. This item/agent must be corresponding to an edge on which they are incident. If an edge item is matched to an edge agent for one round, it cannot generate any social welfare in the other round.

Consequently, in a social welfare maximizing matching, for each edge $e = (x, y, z)$, either the edge items are matched to the edge agent i_e for one round each, giving a combined social welfare of 8, or g_e^1 is matched to x, g_e^2 is matched to y and z is matched to i_e

giving a combined social welfare of 9. For an edge $e = (x, y, z)$, if a repeated matching matches both edge items to the corresponding node agents and the edge agent to the node item z, all for two rounds, we say that the edge gadget is matched completely. If a repeated matching matches both edge items to the edge item for one round each, we say that the edge gadget is matched internally. Any other matching, matches the edge gadget partially.

As any node agent can gain value by being matched to an edge item only, and a node item gives value only to a edge agent, it is enough to see how each edge gadget is matched to calculate the social welfare. Each constructed instance can always generate social welfare $8p$ by matching each of the p edge gadgets internally. To get any additional welfare, some edge gadgets must be matched completely. Each completely matched edge gadget gives social welfare 9. Thus, if a repeated matching has social welfare $8p + k$, it must match k edge gadgets completely. The nodes agents and items of these selected edge gadgets must all be distinct. Thus, the corresponding edges form a 3DM.

Now, if a 3DM M exists, we can match the edge gadgets for the edges in M completely and all other edge gadget internally. All other matches are arbitrary. Thus each gadget generates a social welfare of at least 8 and all the edges in M generate an additional social welfare of 1, for total social welfare of 9 each. Thus, the social welfare of the repeated matching is $8p + k$. □

3.1. Monotone valuations

Fortunately, the problem of computing a repeated matching of maximum social welfare can be solved in polynomial time for monotone valuations, even when the items are mixed. Notice that the instance in the example given at the beginning of Section 3 belongs to the category of monotone non-increasing valuations.

Monotone non-increasing valuations. For this particular case, well-known results on b-matchings can be used to find a social welfare maximizing repeated matching. In the following, we briefly explain how; recall that a b-matching in a bipartite graph is just a subset of the edges that includes at most b edges that are incident to any given node. Gabow and Tarjan [21] show how to compute a maximum-weight b-matching on input an edge-weighted bipartite multigraph in time that is polynomial in b, the size of the graph, and the number of bits required to represent the edge-weights.

Given a repeated matching instance $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$ where each $\{v_i\}_{i\in\mathcal{A}}$ is monotone non-increasing, construct the bipartite multigraph graph $G = (\mathcal{A}, \mathcal{G}, E)$ where E consists of T copies of edge (i, g) for each $i \in \mathcal{A}$ and each $g \in \mathcal{G}$. For each $t \in [T]$, $i \in \mathcal{A}$ and $g \in \mathcal{G}$, we set the edge weight of the t^{th} copy of edge (i, g) to $v_i(g, t)$. Now, since the v_is are monotone non-increasing, we can assume that a maximum-weight T-matching in G has the following *consecutive edge copies* property: if it contains k copies of an edge (i, g), these are the first k copies of weights $v_i(g, 1)$, ..., $v_i(g, k)$. Notice that, if this is not the case, we can redistribute the edge copies of (i, g) between agents appropriately without violating weight maximality. Now, a maximum-weight T-matching M in G naturally defines a repeated matching A_M in I, where each i is matched to each g as many times as the number of copies of edge (i, g) M contains. Furthermore, the social welfare of A_M is equal to the weight of M and can be seen to be optimal. The reason is that any repeated matching corresponds to a T-matching with the consecutive edge copies property.

In Appendix A, we present an alternative approach for finding social welfare maximizing repeated matchings for instances with monotone non-increasing valuations. We use an integer linear program and use an LP solver to compute an extreme solution of the LP relaxation, which, as we show, is guaranteed to be integral.

Monotone non-decreasing valuations. Neither b-matchings nor our linear programming-based approach can be used when all the valuation functions are monotone non-decreasing. Somewhat surprisingly, it suffices to resort to an even simpler ordinary matching computation in this case.

We remark that, on repeated matching instances with constant valuations, there is always a repeated matching of maximum social welfare in which every agent gets the same item in all rounds. To see why, consider any repeated matching A and let t be that round in which the total value the agents get from the items they get in matching A^t is maximum. Then, the repeated matching which uses matching A^t in all rounds has at least as high social welfare with A. Hence, a straightforward maximum-weight matching computation can be used to compute a social welfare maximizing repeated matching for instances with constant valuations. The proof of the next theorem exploits a connection of instances with monotone non-decreasing valuations and instances with constant valuations.

Theorem 2. *Given a repeated instance with monotone non-decreasing valuations, a repeated matching of maximum social welfare can be computed in polynomial time.*

Proof. Consider a repeated matching instance $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$ with monotone non-decreasing valuations. For each agent $i \in \mathcal{A}$, we construct the constant valuation function v_i^c with $v_i^c(g) = \frac{1}{T}\sum_{t=1}^{T} v_i(g, t)$ for each item $g \in \mathcal{G}$. That is, the value that agent i gets from a copy of item g under valuation v_i^c is i's average value from g under v_i in T rounds. Observe that, by the definition of the valuation v_i, $v_i(A_i) \le v_i^c(A_i)$ for any repeated matching A and any agent $i \in \mathcal{A}$. This implies that the social welfare of A under the valuations v_i is not higher than the social welfare under the valuations v_i^c. Hence, the maximum social welfare among all repeated matchings with respect to valuations v_i is not higher than the maximum social welfare among all repeated matchings with respect to valuations v_i^c. Furthermore, the maximum social welfare under v_i^c is achieved by a repeated matching $\widehat{A}$ that uses the same matching in all rounds. Finally, note that $v_i(\widehat{A}_i) = \sum_{t=1}^{T} v_i(g_i, t) = v_i^c(\widehat{A}_i)$, where g_i is the item agent i gets in all rounds under $\widehat{A}$. I.e., the social welfare of $\widehat{A}$ is the same with respect to the original valuations v_i and the modified valuations v_i^c. Thus, to maximize the social welfare, it suffices to compute a single-round matching of maximum social welfare according the valuations v_i^c and repeat it for T rounds. □

4. Computing fair repeated matchings

In this section, we focus on repeated matching instances with goods (i.e., non-negative valuations) and present algorithms that compute EF1 repeated matchings under different conditions. We consider identical valuations in Section 4.1 and conclude with our results for general non-negative valuations in Section 4.2. We first discuss how previous work by Biswas and Barman [11] can be adapted to handle repeated matching instances with constant valuations.

Constant valuations and beyond

We now briefly discuss the seemingly related problem of fair division with cardinality constraints and whether existing results can be used to obtain EF1 repeated matchings in our setting. Biswas and Barman [11] consider an extension of the standard fair division setting where a set of items (goods) needs to be allocated to a set of agents with additive valuations for the items. The additional feature of their problem is that the set of items is partitioned into categories and each category has a cardinality constraint. The objective is now to compute an allocation of the items to the agents, in which the number of items each agent gets from each category does not exceed the cardinality constraint of that category. Biswas and Barman [11] show that allocations that satisfy such cardinality constraints and are furthermore EF1 do exist and can be computed in polynomial time.

Notice that the results of Biswas and Barman [11] can be used to compute EF1 repeated matchings for instances with constant valuations. Indeed, given a repeated matching instance with constant valuations $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$, it suffices to consider a fair division instance I' with the n agents in $\mathcal{A}$, T distinct items for each item g in $\mathcal{G}$, each of value $v_i(g)$ to agent i. We make only one category, with all items belonging to it, with an associated cardinality constraint of T. It can be easily seen that any EF1 allocation for instance I' naturally corresponds to an EF1 repeated matching for instance I and vice versa. Unfortunately, for non-constant valuations, this reduction does not work as it seems impossible to express the history-dependent valuations in our model with additive valuations for items in the model of Biswas and Barman [11].

An algorithmic idea for repeated matchings that we use extensively in the remainder of this section is to begin by assigning $\lfloor T/n \rfloor$ copies of each item to each agent and distribute the remaining $T \bmod n$ copies of each item so that each agent gets at most one additional copy. Can we achieve EF1 in this way for general valuations? This requires the computation of an EF1 repeated matching on instances with $T < n$, in which each agent gets at most one copy of each item (and T copies in total). Even though additivity would not be a problem anymore, it is still not clear how to express such instances in the model of Biswas and Barman [11] using cardinality constraints defined on a *single* partition of the items only.

Furthermore, let us also discuss why a naive implementation of the well-known round-robin allocation fails to achieve EF1 for general repeated matching instances. Consider an instance with two agents 1 and 2, two items g_1 and g_2, and $T = 3$. The valuations for agent 1 and 2 are depicted at the left and right part of the next table, respectively.

item	$T = 1$	$T = 2$	$T = 3$
g_1	2	2	2
g_2	1	10	10

item	$T = 1$	$T = 2$	$T = 3$
g_1	1	10	10
g_2	2	2	2

Allocating the six copies of the items in a round-robin manner to the two agents is done as follows. The agents act one after the other for six steps in total. Whenever it is the turn of an agent to act, she picks an available item copy that maximizes the increase in her valuation. This process will end up with an allocation of the three copies of item g_1 to agent 1 and the three copies of item g_2 to agent 2. Clearly, the valuation of each agent for her bundle is 6, while her valuation for the bundle of the other agent after removing an item copy is 11. Still, elements of round-robin are used by the algorithms we present in the following, but more carefully.

4.1. Identical valuations

Our algorithm for repeated matching instances with identical valuations works as follows. It starts by assigning $\lfloor T/n \rfloor$ copies of each item to each agent. If $T \bmod n > 0$ (i.e., additional copies have to be assigned to the agents so that the repeated matching is correct), the algorithm works in a round robin fashion for $T \bmod n$ phases. In these phases, it uses a fixed ranking of the items according to the value $v(g, \lceil T/n \rceil)$ of their $\lceil T/n \rceil$-th copy. The ranking assigns to each item a distinct integer $\text{rank}(g)$ in $[n]$ such that $\text{rank}(g_1) < \text{rank}(g_2)$ implies that $v(g_1, \lceil T/n \rceil) \geq v(g_2, \lceil T/n \rceil)$. In each round-robin phase, the agents act according to the ordering $1, 2, \ldots, n$. When it is agent i's turn, she picks a copy of the lowest-rank item that is available.

The algorithm appears below as Algorithm 1. It has access to function rank() defined as above and uses the matrix f to store the number of copies of each item an agent gets. The final step is to call routine GenerateFromFreq() to transform f to the repeated matching A; this routine essentially implements the transformation described in Section 2 and is called at the final step of every algorithm we present in the paper.

We now use Algorithm 1 to prove the next statement.

Algorithm 1: Computing an EF1 repeated matching under identical valuations.

```
Input: Identical Valuations Instance $I = \langle \mathcal{A}, \mathcal{G}, v, T \rangle$ with $|\mathcal{A}| = n$
Output: A repeated matching $A$
$f(i,g) \leftarrow \lfloor T/n \rfloor, \forall i \in \mathcal{A}, \forall g \in \mathcal{G}$;
if $T \bmod n > 0$ then
    $x_g \leftarrow T \bmod n, \forall g \in G$;
    for $t = 1$ to $T \bmod n$ do
        for $i = 1$ to $n$ do
            $g' \leftarrow \arg\min_{g : x_g > 0} \text{rank}(g)$;
            $x_{g'} \leftarrow x_{g'} - 1$;
            $f(i,g') \leftarrow \lceil T/n \rceil$;
$A \leftarrow \text{GenerateFromFreq}(f)$;
```

Theorem 3. *Given a repeated matching instance with identical non-negative valuations, an EF1 repeated matching exists and can be computed in polynomial time.*

Proof. Algorithm 1 clearly runs in polynomial time. It remains to prove that it always returns an EF1 repeated matching. Consider its application to a repeated matching instance $I = \langle \mathcal{A}, \mathcal{G}, v, T \rangle$, where v is non-negative. The repeated matching returned is clearly EF1 if T is an integer multiple of n; in this case, all agents get the same number of copies of all items and nobody is envious.

Otherwise, since $T \bmod n \leq n-1$ copies of each item are available in the round-robin phases and all the remaining $T \bmod n$ copies of each item are picked in consecutive round-robin steps, no agent gets more than one copy of the same item in the round robin phases. Let $g_{i,t}$ be the item agent i gets in the round robin phase $t \in \{1,2,...,T \bmod n\}$. Consider two agents i and j and observe that the repeated matching A returned by Algorithm 1 satisfies

$$\begin{aligned}
& v(A_i) - v(A_j \setminus \{g_{j,1}\}) \\
&= \sum_{t=1}^{(T \bmod n)-1} (v(g_{i,t}, \lceil T/n \rceil) - v(g_{j,t+1}, \lceil T/n \rceil)) + v(g_{i,T \bmod n}) \\
&\geq \sum_{t=1}^{(T \bmod n)-1} (v(g_{i,t}, \lceil T/n \rceil) - v(g_{j,t+1}, \lceil T/n \rceil)) \geq 0,
\end{aligned}$$

as EF1 requires. The equality follows since both agents i and j get $\lfloor T/n \rfloor$ copies of each item at the beginning of the algorithm and, then, the valuation difference is due to the $\lceil T/n \rceil$-th copies of items allocated in the round-robin phases. The first inequality is due to the non-negativity of valuations. The second one follows since the item that agent i picks at the round-robin phase t has not higher rank than the item agent j picks in the next phase $t+1$. □

4.2. General valuations

We now prove that EF1 repeated matchings can be computed in polynomial time for general non-negative valuations when the number T of rounds and the number n of agents/items satisfy a particular condition.

Theorem 4. *Given a repeated matching instance I with n agents/goods and T rounds such that $T \bmod n \in \{0,1,2,n-1\}$, an EF1 repeated matching exists and can be computed in polynomial time.*

We prove Theorem 4 constructively, by defining two algorithms for the cases $T \bmod n \in \{0,1,2\}$ (Algorithm 2) and $T \bmod n = n-1$ (Algorithm 3).

Algorithm 2 computes the number of copies of each item that each agent gets as follows. First, it gives to each agent $\lfloor T/n \rfloor$ copies of each item (line 1). If $T \bmod n \neq 0$, it then runs a round-robin phase (lines 2-7) and then, if $T \bmod n = 2$, it runs an additional reverse round-robin phase (lines 8-13). In the round-robin phase, the agents act according to the ordering $1,2,...,n$ (see the for-loop in lines 4-7). When it is agent i's turn to act, she gets the item $\hat{g}$ (identified in line 5) for which her value for the $\lceil T/n \rceil$-th copy is maximum among the items that have not been given to agents who acted before i in the round-robin phase (the set variable P is used to identify these items). In the reverse round-robin phase, the agents act according to the ordering $n, n-1, ..., 1$ (see the for-loop in lines 10-13). When it is agent i's turn to act, she gets the item $\hat{g}$ (identified in line 11) for which her value for the next copy is maximum among the items that have not been given to agents who acted before i in the reverse round-robin phase. Finally, the algorithm transforms the matrix f indicating the number of copies of each item each agent gets to a repeated matching by calling routine GenerateFromFreq(). Algorithm 2 clearly runs in polynomial time.

Lemma 1. *The repeated matching $A = (A_1, ..., A_n)$ produced by Algorithm 2 is* EF1.

Algorithm 2: Computing an EF1 repeated matching.

```
Input: Instance $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$ with $|\mathcal{A}| = n$ and $T \bmod n \in \{0,1,2\}$
Output: A repeated matching $A$
$f(i,g) \leftarrow \lfloor T/n \rfloor, \forall i \in \mathcal{A}, \forall g \in G$;
if $T \bmod n > 0$ then
    $P \leftarrow \mathcal{G}$;
    for $i = 1$ to $n$ do
        $\hat{g} \leftarrow \arg\max_{g\in P} v_i(g, \lceil T/n \rceil)$;
        $f(i,\hat{g}) \leftarrow \lceil T/n \rceil$;
        $P \leftarrow P \setminus \{\hat{g}\}$;
if $T \bmod n = 2$ then
    $P \leftarrow \mathcal{G}$;
    for $i = n$ to $1$ do
        $\hat{g} \leftarrow \arg\max_{g\in P} v_i(g, f(i,g)+1)$;
        $f(i,\hat{g}) \leftarrow f(i,\hat{g}) + 1$;
        $P \leftarrow P \setminus \{\hat{g}\}$;
$A \leftarrow$ GenerateFromFreq($f$);
```

Proof. Let S denote the multiset that contains each item with multiplicity $\lfloor T/n \rfloor$. If $T \bmod n = 0$, then $A_i = S$ for every agent i and, hence, agents are not envious of each other. If $T \bmod n = 1$, the final repeated matching is obtained after the execution of the round-robin phase. Consider two agents i and j. Denoting by g_j the item agent j gets in this phase, agent i has value $v_i(A_i) \geq v_i(S) = v_i(A_j \setminus \{g_j\})$, i.e., she satisfies the EF1 condition.

If $T \bmod n = 2$, the final repeated matching is obtained after the execution of the reverse round-robin phase. Consider two agents i and j with $i < j$. Let g_i^1 and g_j^1 be the items the agents i and j get in the round-robin phase and g_i^2 and g_j^2 be the items they get in the reverse round-robin phase, respectively. Agent i has value

$$\begin{aligned} v_i(A_i) &\geq v_i(S) + v_i(g_i^1, \lceil T/n \rceil) \\ &\geq v_i(S) + v_i(g_j^1, \lceil T/n \rceil) = v_i(A_j \setminus \{g_j^2\}). \end{aligned}$$

The second inequality follows since agent i prefers item g_i^1 to item g_j^1 in the round-robin phase. For agent j, we distinguish between two cases. Let μ denote the multiplicity of item g_j^2 in A_j. If $g_j^1 \neq g_i^2$, we have that, in the reverse round-robin phase, agent j prefers the μ-th copy of g_j^2 to the $\lceil T/n \rceil$-th copy of g_i^2, i.e., $v_j(g_j^2, \mu) \geq v_j(g_i^2, \lceil T/n \rceil)$. Then, we have

$$\begin{aligned} v_j(A_j) &\geq v_j(S) + v_j(g_j^2, \mu) \\ &\geq v_j(S) + v_j(g_i^2, \lceil T/n \rceil) = v_j(A_i \setminus \{g_i^1\}). \end{aligned}$$

If $g_j^1 = g_i^2$, we have

$$\begin{aligned} v_j(A_j) &\geq v_j(S) + v_j(g_j^1, \lceil T/n \rceil) \\ &= v_j(S) + v_j(g_i^2, \lceil T/n \rceil) = v_j(A_i \setminus \{g_i^1\}). \end{aligned}$$

Thus, the EF1 conditions for agents i and j are satisfied. □

Algorithm 3 uses a similar structure. It starts by giving $\lceil T/n \rceil$ copies of each item to each agent (in line 1) and then removes the copy of a distinct item from each agent by running a round-robin phase (lines 2-6). When it is agent i's turn to act, she gets rid of a copy of the item $\hat{g}$ (identified in line 4) for which her value for the $\lceil T/n \rceil$-th copy is minimum among the items that have not been gotten rid by agents who acted before i in the round-robin phase.

Algorithm 3: Computing an EF1 repeated matching.

```
  Input: Instance $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$ with $|\mathcal{A}| = n$ and $T \bmod n = n - 1$
  Output: A repeated matching $A$
1 $f(i,g) \leftarrow \lceil T/n \rceil, \forall i \in \mathcal{A}, \forall g \in G$;
2 $P \leftarrow \mathcal{G}$;
3 for $i = 1$ to $n$ do
4     $\hat{g} \leftarrow \arg\min_{g\in P} v_i(g, \lceil T/n \rceil)$;
5     $f(i,\hat{g}) \leftarrow f(i,\hat{g}) - 1$;
6     $P \leftarrow P \setminus \{\hat{g}\}$;
7 $A \leftarrow$ GenerateFromFreq($f$);
```

Lemma 2. *The repeated matching $A = (A_1, ..., A_n)$ produced by Algorithm 3 is* EF1.

Proof. Let i and j be two agents and denote by g_i and g_j the items that are removed from their bundles in the round-robin phase. We have

$$v_i(A_i) = v_i(A_j) + v_i(g_j, \lceil T/n \rceil) - v_i(g_i, \lceil T/n \rceil)$$
$$\geq v_i(A_j) - v_i(g_i, \lceil T/n \rceil) = v_i(A_j \setminus \{g_i\})$$

as desired. The last equality follows since A_j has exactly $\lceil T/n \rceil$ copies of item g_i. □

Theorem 4 implies the following corollary.

Corollary 1. *In repeated matching instances with up to four agents/goods, an* EF1 *repeated matching always exists.*

5. Are fairness and efficiency compatible?

In this section, we show that achieving the concepts of efficiency and fairness simultaneously is computationally intractable. In particular, we show in Theorem 5 below that even approximating the maximum social welfare of EF1 repeated matching is hard. Our proof is inspired by a reduction by Barman et al. [5] but is more involved. Interestingly, it uses instances with constant valuations and comes in sharp contrast to achieving the two concepts separately. For such instances, an EF1 repeated matching can be computed in polynomial time by the techniques of Biswas and Barman [11] (see the related discussion at the beginning of Section 4), while a polynomial time algorithm for computing social welfare maximizing repeated matchings follows by Theorem 2.

Theorem 5. *For every constant $\epsilon > 0$, approximating the maximum social welfare of* EF1 *repeated matchings on instances with n agents/goods and T rounds within a factor of $O\left(\min\{n^{1/3-\epsilon}, T^{1-\epsilon}\}\right)$ is NP-hard.*

Proof. We present a polynomial-time reduction, which, given a graph $G = (V, E)$, constructs a repeated matching instance $I(G)$ in which the maximum social welfare over EF1 repeated matchings is in $[K, K+1)$ if and only if the maximum independent set in graph G has size K. Our construction leads to instances with $n \leq |V|^3$ agents/items and $T = |V|$ rounds. Then, the theorem follows by the next well-known result by Zuckerman [33].

Theorem 6 *(Zuckerman [33]). For every constant $\epsilon > 0$, approximating the maximum independent set of a graph $G = (V, E)$ within a factor of $|V|^{1-\epsilon}$ is NP-hard.*

Let δ be such that $0 < \delta < |V|^{-2}$. Let $G = (V, E)$ be a graph. Without loss of generality, we can assume that G has no isolated nodes, as the existence of such nodes just makes the independent set problem easier. Given graph $G = (V, E)$, the instance $I(G)$ has $T = |V|$ rounds and $n = (2|V| + 1)|E| + 1$ agents/items. For every edge $e \in E$, $I(G)$ has $2|V| + 1$ *edge agents* identified as (e, i) for $i = 1, 2, ..., 2|V| + 1$. There is also a *special agent* s. For every node $u \in V$, there is a *node item* g_u. The instance also has $n - |V|$ *dummy items*. For edge $e = (x, y) \in E$, $i \in [2|V| + 1]$, and $t \in [T]$, the valuation of the edge agent (e, i) for the t^{th} copy of the node item g_u is $v_{e,i}(g_u, t) = \delta$ if $u = x$ or $u = y$, and $v_{e,i}(g_u, t) = 0$ otherwise. For node $u \in V$ and $t \in [T]$, the valuation of the special agent for the t^{th} copy of the node item g_u is $v_s(g_u, t) = 1$. All agents have zero valuations for the dummy items.

Let K be the size of the maximum independent set in G. We claim that any EF1 repeated matching of $I(G)$ has social welfare less than $K + 1$. This will follow by two observations for any EF1 repeated matching A. First, for every edge e, there is some $i \in [2|V| + 1]$ such that the edge agent (e, i) has value 0. Assume that this is not true for edge $e = (x, y)$. Hence, $2|V| + 1$ copies of the node items g_x and g_y have been given to the edge agents corresponding to edge e. However, we only have $|V|$ copies of each item. Second, consider the node items the special agent gets. As for each edge $e = (x, y)$, there is some agent (e, i) who has zero value, the special agent can get at most one copy of node items g_x or g_y. As this holds for every $e \in E$, the node items that the special agent gets correspond to the nodes in an independent set in G. Hence, her value is at most K. The total value the edge agents get from the $|V|$ node items they get is at most $|V|^2 \cdot \delta < 1$. Hence, the social welfare of repeated matching A is less than $K + 1$.

We now show that an EF1 repeated matching of social welfare in $[K, K+1)$ does exist, when the graph G has an independent set S of size K. First, the special agent gets a single copy of node item g_x for each $x \in S$. The remaining copies of the node items are given to the edge agents in such a way that each edge agent corresponding to edge $e = (x, y)$ gets at most one copy of either g_x or g_y. This is always possible, since for every edge $e = (x, y)$, there are $2|V| + 1$ edge agents to get at most one copy of either node item g_x or node item g_y. Then, the copies of the dummy items are distributed so that each agent has exactly $|V|$ item copies. As every edge agent has at most one copy of a node item, the EF1 conditions between any two of them are satisfied. Finally, the EF1 is satisfied between any edge agent and the special agent since the special agent gets at most one item copy for which the edge agent has positive value. □

6. Swap envy-freeness

We now specifically turn our attention to repeated matching instances with mixed items. Consider the following instance with $n = 2$ and $T = 1$. One of the items is a good and the other is a chore. There are exactly two possible matchings. In either, the classical extension of EF1 for mixed items from the fair division literature (e.g., see [3]), which requires that the value of an agent is higher than that of another either by removing a single item from either one of the two bundles, is not satisfied. Motivated by this simple example, we propose and investigate an alternate notion of fairness to EF1 for repeated matchings, which we call *swap envy-freeness* (swapEF).

Definition 4 *(*swapEF*).* Let $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$ be a repeated matching instance with mixed items. A repeated matching $A = (A_1, ..., A_n)$ in I is swapEF if for every pair of agents $i, j \in \mathcal{A}$, either (i) or (ii) is true:

(i) $v_i(A_i) \geq v_i(A_j)$;
(ii) There exist items $g_i \in A_i$ and $g_j \in A_j$ such that $v_i(A_i \cup \{g_j\} \setminus \{g_i\}) \geq v_i(A_j \cup \{g_i\} \setminus \{g_j\})$.

Condition (ii) requires that the value agent i has for her bundle A_i after replacing a copy of item g_i with an extra copy of item g_j is at least as high as her value for the bundle A_j of agent j after exchanging a copy of item g_j with a copy of item g_i. For instance, in the example of one good and one chore with $T = 1$, we find that while EF1 cannot be satisfied, each of the two repeated matchings are actually swapEF. This is because the agent who is matched to the chore will envy the other agent, but should the two be swapped, this envy would be mitigated. Now if $T = 2k + 1$ in this example with valuations staying constant with time, any repeated matching that matched one agent to the chore for $k + 1$ rounds will be swapEF. The agent matched to the chore for $k + 1$ rounds, will envy the other agent, but exchanging just one copy of the chore for one of the good would mitigate this envy.

We first find that Algorithm 1 successfully finds a swapEF repeated matching, even without the non-negativity constraint on valuations (the rank definition can be trivially adapted).

Lemma 3. *Given a repeated matching instance $I = \langle \mathcal{A}, \mathcal{G}, v, T\rangle$ with identical valuations, the repeated matching returned by Algorithm 1 is* swapEF.

Proof. First observe that if T is an integer multiple of n, the repeated matching computed by Algorithm 1 creates no envy to any agent and, hence, it is swapEF as well. Now, assume that T is not an integer multiple of n; the algorithm will execute $T \bmod n$ round-robin phases in this case. Denote by $g_{i,t}$ the item agent i gets in the round-robin phase $t \in \{1, 2, ..., T \bmod n\}$. Agent i gets $\lceil T/n\rceil$ copies of each of these items, while it uses only $\lfloor T/n\rfloor$ copies of the rest. Then, for any pair of agents i and j, observe that

$$v(A_i) - v(A_j) = \sum_{t=1}^{T \bmod n} (v(g_{i,t}, \lceil T/n\rceil) - v(g_{j,t}, \lceil T/n\rceil)).$$

If $i < j$, then it is also $v(g_{i,t}, \lceil T/n\rceil) \geq v(g_{j,t}, \lceil T/n\rceil)$ for every round-robin phase t, which implies that $v(A_i) \geq v(A_j)$. The inequality is clear if both agents i and j get a copy of the same item in phase t. If this is not the case, the item agent i picks has lower rank than the item agent j picks later. This implies that $v(g_{i,t}, \lceil T/n\rceil) \geq v(g_{j,t}, \lceil T/n\rceil)$, too.

Now assume that $i > j$. By the argument above, we also get $v(A_i) \geq v(A_j)$ when agents i and j get a copy of the same item in each round. So, in the following, let us assume that this is not the case and denote by t_1 and t_2 the first and the last round-robin phase in which agents i and j get different items. Then,

$$\begin{aligned}
& v(A_i \cup \{g_{j,t_1}\} \setminus \{g_{i,t_2}\}) - v(A_j \cup \{g_{i,t_2}\} \setminus \{g_{j,t_1}\}) \\
&= \sum_{t=1}^{t_1-1} (v(g_{i,t}, \lceil T/n\rceil) - v(g_{j,t}, \lceil T/n\rceil)) \\
&+ v(g_{j,t_1}, \lceil T/n\rceil) + \sum_{t=t_1}^{t_2-1} (v(g_{i,t}, \lceil T/n\rceil) - v(g_{j,t+1}, \lceil T/n\rceil)) \\
&- v(g_{i,t_2}, \lceil T/n\rceil) + \sum_{t=t_2+1}^{T \bmod n} (v(g_{i,t}, \lceil T/n\rceil) - v(g_{j,t}, \lceil T/n\rceil)) \\
&\geq v(g_{j,t_1}, \lceil T/n\rceil) - v(g_{i,t_2}, \lceil T/n\rceil) \geq 0.
\end{aligned}$$

The first inequality follows since the first and third sums are equal to 0 and the second one is non-negative. This is due to the following observations. First, notice that, by definition, both agents i and j get a copy of the same item in phases from 1 to $t_1 - 1$ and from $t_2 + 1$ to $T \bmod n$. Second, notice that the item $g_{i,t}$ that agent i picks in round-robin phase t is either the same with the one that agent j picks in the next round-robin phase $t + 1$ or one that has lower rank (and, thus, is at least as preferable). The second inequality is due to the fact that the item that agent j picks in the round-robin phase t_1 is at least as preferable to the one agent i picks later in the round-robin phase $t_2 \geq t_1$.

We have established the swapEF requirements in any case, and the proof is complete. □

We now turn our attention to general valuations.

Theorem 7. *Given a repeated matching instance I with mixed items, n agents, and T rounds such that $T \bmod n \in \{0, 1, 2, n-2, n-1\}$, a* swapEF *repeated matching exists and can be computed in polynomial time.*

The proof of Theorem 7 uses Algorithm 2 from Section 4 for instances with $T \bmod n \in \{0, 1, 2\}$. For instances with $T \bmod n \in \{n-2, n-1\}$, we use an extension of Algorithm 3 from Section 4, which runs an additional reverse round robin phase to remove one more distinct item from each agent when $T \bmod n = n-2$. We refer to this as Algorithm 4; the lines 7-12 implement the reverse round-robin phase, while the lines 1-6 are identical to Algorithm 3.

Algorithm 4: Computing a swapEF repeated matching.

```
   Input: Instance $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$ with $|\mathcal{A}| = n$ and $T \bmod n \in \{n-1, n-2\}$
   Output: A repeated matching $A$
1  $f(i,g) \leftarrow \lceil T/n \rceil, \forall i \in \mathcal{A}, \forall g \in G$;
2  $P \leftarrow \mathcal{G}$;
3  for $i = 1$ to $n$ do
4      $\hat{g} \leftarrow \arg\min_{g\in P} v_i(g, \lceil T/n \rceil)$;
5      $f(i,\hat{g}) \leftarrow f(i,\hat{g}) - 1$;
6      $P \leftarrow P \setminus \{\hat{g}\}$;
7  if $T \bmod n = n-2$ then
8      $P \leftarrow \mathcal{G}$;
9      for $i = n$ to $1$ do
10         $\hat{g} \leftarrow \arg\min_{g\in P} v_i(g, f(i,g))$;
11         $f(i,\hat{g}) \leftarrow f(i,\hat{g}) - 1$;
12         $P \leftarrow P \setminus \{\hat{g}\}$;
13 $A \leftarrow$ GenerateFromFreq($f$);
```

The properties of Algorithms 2 and 4 regarding swapEF are given by the next two lemmas, which, together with the fact that both algorithms run in polynomial time, complete the proof of Theorem 7.

Lemma 4. *The repeated matching $A = (A_1, ..., A_n)$ produced by Algorithm 2 is* swapEF.

Proof. Let S denote the multiset that contains each item with multiplicity $\lfloor T/n \rfloor$. If $T \bmod n = 0$, then $A_i = S$ for every agent i and, hence, the agents are not envious of each other. If $T \bmod n = 1$, the final repeated matching is obtained after the execution of the round-robin phase. Consider two agents i and j and let g_i and g_j be the items the two agents get in this phase, respectively. If $v_i(g_i, \lceil T/n \rceil) \geq v_i(g_j, \lceil T/n \rceil)$, then

$$v_i(A_i) = v_i(S) + v_i(g_i, \lceil T/n \rceil) \geq v_i(S) + v_i(g_j, \lceil T/n \rceil) = v_i(A_j).$$

Otherwise, if $v_i(g_i, \lceil T/n \rceil) < v_i(g_j, \lceil T/n \rceil)$, then

$$\begin{aligned} v_i(A_i \cup \{g_j\} \setminus \{g_i\}) &= v_i(S) + v_i(g_j, \lceil T/n \rceil) \\ &> v_i(S) + v_i(g_i, \lceil T/n \rceil) \\ &= v_i(A_j \cup \{g_i\} \setminus \{g_j\}). \end{aligned}$$

In both cases, the swapEF conditions are satisfied.

If $T \bmod n = 2$, the final repeated matching is obtained after the execution of the reverse round-robin phase. Consider two agents i and j. Let g_i^1 and g_j^1 be the items agents i and j get in the round-robin phase and g_i^2 and g_j^2 be the items they get in the reverse round-robin phase, respectively. We distinguish between three cases. If $|\{g_i^1, g_i^2\} \cap \{g_j^1, g_j^2\}| = 2$, then A_i and A_j are effectively identical and agent i does not envy agent j. If $|\{g_i^1, g_i^2\} \cap \{g_j^1, g_j^2\}| = 1$, assume, without loss of generality, that $g_i^1 = g_j^2 = g$ and observe that A_i has $\lceil T/n \rceil$ copies of g_i^2 and $\lfloor T/n \rfloor$ copies of g_j^1 and A_j has $\lceil T/n \rceil$ copies of g_j^1 and $\lfloor T/n \rfloor$ copies of g_i^2. If $v_i(g_i^2, \lceil T/n \rceil) \geq v_i(g_j^1, \lceil T/n \rceil)$, then

$$\begin{aligned} v_i(A_i) &= v_i(S \cup \{g\}) + v_i(g_i^2, \lceil T/n \rceil) \\ &\geq v_i(S \cup \{g\}) + v_i(g_j^1, \lceil T/n \rceil) = v_i(A_j). \end{aligned}$$

Otherwise, if $v_i(g_i^2, \lceil T/n \rceil) < v_i(g_j^1, \lceil T/n \rceil)$, then

$$\begin{aligned}v_i(A_i \cup \{g_j^1\} \setminus \{g_i^2\}) &= v_i(S) + v_i(g_j^1, \lceil T/n \rceil)\\ &> v_i(S) + v_i(g_i^2, \lceil T/n \rceil)\\ &= v_i(A_j \cup \{g_i^2\} \setminus \{g_j^1\}).\end{aligned}$$

So, the swapEF conditions are satisfied.

It remains to consider the case where g_i^1, g_i^2, g_j^1, and g_j^2 are distinct. Then, A_i contains $\lceil T/n \rceil$ copies of g_i^1 and g_i^2 and $\lfloor T/n \rfloor$ copies of g_j^1 and g_j^2 and A_j contains $\lceil T/n \rceil$ copies of g_j^1 and g_j^2 and $\lfloor T/n \rfloor$ copies of g_i^1 and g_i^2. If $i < j$, agent i acts before agent j in the round-robin phase and, hence, $v_i(g_i^1, \lceil T/n \rceil) \geq v_i(g_j^1, \lceil T/n \rceil)$. If $v_i(g_i^2, \lceil T/n \rceil) \geq v_i(g_j^2, \lceil T/n \rceil)$, then

$$\begin{aligned}v_i(A_i) &= v_i(S) + v_i(g_i^1, \lceil T/n \rceil) + v_i(g_i^2, \lceil T/n \rceil)\\ &\geq v_i(S) + v_i(g_j^1, \lceil T/n \rceil) + v_i(g_j^2, \lceil T/n \rceil) = v_i(A_j),\end{aligned}$$

and agent i does not envy agent j. Otherwise, if $v_i(g_i^2, \lceil T/n \rceil) < v_i(g_j^2, \lceil T/n \rceil)$, then

$$\begin{aligned}v_i(A_i \cup \{g_j^2\} \setminus \{g_i^2\}) &= v_i(S) + v_i(g_i^1, \lceil T/n \rceil) + v_i(g_j^2, \lceil T/n \rceil)\\ &> v_i(S) + v_i(g_j^1, \lceil T/n \rceil) + v_i(g_i^2, \lceil T/n \rceil)\\ &= v_i(A_i \cup \{g_i^2\} \setminus \{g_j^2\}),\end{aligned}$$

and the swapEF condition is satisfied. If $i > j$, agent i acts before j in the reverse round-robin phase and, hence, $v_i(g_i^2, \lceil T/n \rceil) \geq v_i(g_j^2, \lceil T/n \rceil)$. If $v_i(g_i^1, \lceil T/n \rceil) \geq v_i(g_j^1, \lceil T/n \rceil)$, then

$$\begin{aligned}v_i(A_i) &= v_i(S) + v_i(g_i^1, \lceil T/n \rceil) + v_i(g_i^2, \lceil T/n \rceil)\\ &\geq v_i(S) + v_i(g_j^1, \lceil T/n \rceil) + v_i(g_j^2, \lceil T/n \rceil) = v_i(A_j),\end{aligned}$$

and agent i does not envy agent j. Otherwise, if $v_i(g_i^1, \lceil T/n \rceil) < v_i(g_j^1, \lceil T/n \rceil)$, then

$$\begin{aligned}v_i(A_i \cup \{g_j^1\} \setminus \{g_i^1\}) &= v_i(S) + v_i(g_i^2, \lceil T/n \rceil) + v_i(g_j^1, \lceil T/n \rceil)\\ &> v_i(S) + v_i(g_j^2, \lceil T/n \rceil) + v_i(g_i^1, \lceil T/n \rceil)\\ &= v_i(A_i \cup \{g_i^1\} \setminus \{g_j^1\}),\end{aligned}$$

and the swapEF condition is again satisfied. □

Lemma 5. *The repeated matching $A = (A_1, ..., A_n)$ produced by Algorithm 4 is* swapEF.

Proof. Let S denote the multiset that contains each item with multiplicity $\lceil T/n \rceil$. We first consider the case where $T \bmod n = n - 1$. Let i and j be two agents and denote by g_i and g_j the items that are removed from their bundles in the round-robin phase. If $v_i(g_i, \lceil T/n \rceil) \geq v_i(g_j, \lceil T/n \rceil)$, then

$$\begin{aligned}v_i(A_i \cup \{g_i\} \setminus \{g_j\}) &= v_i(S) - v_i(g_j, \lceil T/n \rceil)\\ &\geq v_i(S) - v_i(g_i, \lceil T/n \rceil) = v_i(A_j \cup \{g_j\} \setminus \{g_i\}).\end{aligned}$$

Otherwise, if $v_i(g_i, \lceil T/n \rceil) < v_i(g_j, \lceil T/n \rceil)$, then

$$v_i(A_i) = v_i(S) - v_i(g_i, \lceil T/n \rceil) > v_i(S) - v_i(g_j, \lceil T/n \rceil) = v_i(A_j).$$

Thus, swapEF is satisfied by Algorithm 4 whenever $t \bmod n = n - 1$. If $T \bmod n = n - 2$, the final repeated matching is obtained after the execution of the reverse round-robin phase. Consider two agents i and j. Let g_i^1 and g_j^1 be the items agents i and j remove in the round-robin phase and g_i^2 and g_j^2 be the items they remove in the reverse round-robin phase, respectively. We distinguish between three cases. If $|\{g_i^1, g_i^2\} \cap \{g_j^1, g_j^2\}| = 2$, then A_i and A_j are identical and agent i does not envy agent j. If $|\{g_i^1, g_i^2\} \cap \{g_j^1, g_j^2\}| = 1$, assume, without loss of generality, that $g_i^1 = g_j^2 = g$ and observe that A_i has $\lfloor T/n \rfloor$ copies of g_i^2 and $\lceil T/n \rceil$ copies of g_j^1 and A_j has $\lfloor T/n \rfloor$ copies of g_j^1 and $\lceil T/n \rceil$ copies of g_i^2. If $v_i(g_i^2, \lceil T/n \rceil) \leq v_i(g_j^1, \lceil T/n \rceil)$, then

$$\begin{aligned}v_i(A_i) &= v_i(S \setminus \{g\}) - v_i(g_i^2, \lceil T/n \rceil)\\ &\geq v_i(S \setminus \{g\}) - v_i(g_j^1, \lceil T/n \rceil) = v_i(A_j).\end{aligned}$$

Otherwise, if $v_i(g_i^2, \lceil T/n \rceil) > v_i(g_j^1, \lceil T/n \rceil)$, then

$$v_i(A_i \cup \{g_i^2\} \setminus \{g_j^1\}) = v_i(S) - v_i(g, \lceil T/n \rceil) - v_i(g_j^1, \lceil T/n \rceil)$$

$$\begin{aligned} &> v_i(S) - v_i(g, \lceil T/n \rceil) - v_i(g_i^2, \lceil T/n \rceil) \\ &= v_i(A_j \cup \{g_j^1\} \setminus \{g_i^2\}). \end{aligned}$$

So, the swapEF conditions are satisfied in this case.

It remains to consider the case where g_i^1, g_i^2, g_j^1, and g_j^2 are distinct. Then, A_i contains $\lfloor T/n \rfloor$ copies of g_i^1 and g_i^2 and $\lceil T/n \rceil$ copies of g_j^1 and g_j^2 and A_j contains $\lfloor T/n \rfloor$ copies of g_j^1 and g_j^2 and $\lceil T/n \rceil$ copies of g_i^1 and g_i^2. If $i < j$, agent i acts before agent j in the round-robin phase and, hence, $v_i(g_i^1, \lceil T/n \rceil) \leq v_i(g_j^1, \lceil T/n \rceil)$. If $v_i(g_i^2, \lceil T/n \rceil) \leq v_i(g_j^2, \lceil T/n \rceil)$, then

$$\begin{aligned} v_i(A_i) &= v_i(S) - v_i(g_i^1, \lceil T/n \rceil) - v_i(g_i^2, \lceil T/n \rceil) \\ &\geq v_i(S) - v_i(g_j^1, \lceil T/n \rceil) - v_i(g_j^2, \lceil T/n \rceil) = v_i(A_j), \end{aligned}$$

and agent i does not envy agent j. Otherwise, if $v_i(g_i^2, \lceil T/n \rceil) > v_i(g_j^2, \lceil T/n \rceil)$, then

$$\begin{aligned} v_i(A_i \cup \{g_i^2\} \setminus \{g_j^2\}) &= v_i(S) - v_i(g_i^1, \lceil T/n \rceil) - v_i(g_j^2, \lceil T/n \rceil) \\ &> v_i(S) - v_i(g_j^1, \lceil T/n \rceil) - v_i(g_i^2, \lceil T/n \rceil) \\ &= v_i(A_i \cup \{g_j^2\} \setminus \{g_i^2\}), \end{aligned}$$

and the swapEF condition is satisfied. If $i > j$, agent i acts before j in the reverse round-robin phase and, hence, $v_i(g_i^2, \lceil T/n \rceil) \leq v_i(g_j^2, \lceil T/n \rceil)$. If $v_i(g_i^1, \lceil T/n \rceil) \leq v_i(g_j^1, \lceil T/n \rceil)$, then

$$\begin{aligned} v_i(A_i) &= v_i(S) - v_i(g_i^1, \lceil T/n \rceil) - v_i(g_i^2, \lceil T/n \rceil) \\ &\geq v_i(S) - v_i(g_j^1, \lceil T/n \rceil) - v_i(g_j^2, \lceil T/n \rceil) = v_i(A_j), \end{aligned}$$

and agent i does not envy agent j. Otherwise, if $v_i(g_i^1, \lceil T/n \rceil) > v_i(g_j^1, \lceil T/n \rceil)$, then

$$\begin{aligned} v_i(A_i \cup \{g_i^1\} \setminus \{g_j^1\}) &= v_i(S) - v_i(g_i^2, \lceil T/n \rceil) - v_i(g_j^1, \lceil T/n \rceil) \\ &> v_i(S) - v_i(g_j^2, \lceil T/n \rceil) - v_i(g_i^1, \lceil T/n \rceil) \\ &= v_i(A_i \cup \{g_j^1\} \setminus \{g_i^1\}), \end{aligned}$$

and the swapEF condition is again satisfied. □

Theorem 7 implies the following corollary.

Corollary 2. *In repeated matching instances with mixed items and up to five agents/items, a* swapEF *repeated matching always exists.*

We conclude this section with a comparison of EF1 and swapEF. While the two fairness notions have similar definitions, they are actually incomparable. Clearly, swapEF does not imply EF1 as it is trivially satisfied in the simple motivating example with one good and one chore presented at the beginning of this section. However, given that we use largely the same algorithms for swapEF as we did for EF1, one may believe intuitively that for goods alone, EF1 implies swapEF. This is not the case though. Consider an instance with three rounds and two agents with identical constant valuations $v(1,t) = 3$ and $v(2,t) = 2$ for two items. Giving item 1 to one agent and item 2 to the other for all three rounds is EF1 but not swapEF.

7. Open problems

Our work leaves several interesting open problems that deserve investigation. Understanding social welfare maximization is the first one. Is the problem hard for instances with two rounds? Recall that our hardness reduction in the proof of Theorem 1 uses three rounds while the problem is in P for a single round. What about approximation algorithms when the items are goods and valuations are not necessarily monotone? Is a constant approximation ratio possible? Regarding fairness, the most important open question is whether EF1 repeated matchings exist for any instance with goods. Furthermore, is EF1 compatible with different notions of efficiency than the utilitarian social welfare we have used here? For example, what about the egalitarian or Nash social welfare? Is EF1 compatible with Pareto-efficiency? For instances with mixed items, do swapEF repeated matchings always exist? Again, how do they interplay with Pareto-efficiency? In general, swapEF deserves investigation in other fair division settings that involve mixed items.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A. Social welfare maximization under monotone non-increasing valuations

We now give an alternative proof to Theorem 8 below. This is not a new result; we remind the reader that the proof already follows by well-known results on weighted bipartite b-matchings (see the discussion in Section 3.1).

Theorem 8. *Given a repeated instance with monotone non-increasing valuations, a repeated matching of maximum social welfare can be computed in polynomial time.*

Proof. Consider a repeated matching instance $I = \langle \mathcal{A}, \mathcal{G}, \{v_i\}_{i\in\mathcal{A}}, T\rangle$. We express the problem of computing a repeated matching of maximum social welfare as the following integer linear program:

$$\begin{aligned}
\max \quad & \sum_{i\in\mathcal{A}}\sum_{g\in\mathcal{G}}\sum_{t=1}^{T} x_{i,g,t}\cdot v_i(g,t) & \\
\text{s.t.:} \quad & \sum_{g\in\mathcal{G},t\in[T]} x_{i,g,t} = T, & \forall i\in\mathcal{A}\\
& \sum_{i\in\mathcal{A},t\in[T]} x_{i,g,t} = T, & \forall g\in\mathcal{G}\\
& x_{i,g,t} \geq x_{i,g,t+1}, & \forall i\in\mathcal{A}, g\in G, t\in[T-1]\\
& x_{i,g,t}\in\{0,1\}, & \forall i\in\mathcal{A},\ g\in\mathcal{G},\ t\in[T]
\end{aligned}$$

The binary indicator variable $x_{i,g,t}$ denotes whether agent i gets the t^{th} copy of item g ($x_{i,g,t}=1$) or not ($x_{i,g,t}=0$). Then, the objective is clearly to maximize the social welfare, the total value the agents get from their copies of items. The first set of constraints requires that each agent gets exactly T copies of the items in the T rounds. The second one requires that each item is assigned in all the T rounds. The third one ensures that an agent can get her $(t+1)^{\text{th}}$ copy of an item only after she gets the t^{th} copy.

We now relax the integrality constraint by replacing $x_{i,g,t}\in\{0,1\}$ with $x_{i,g,t}\in[0,1]$. In this way, we get a linear program (LP). Well-known solvers, implementing variants of the *ellipsoid method* [25,32], can solve this LP in polynomial time and compute an *extreme* solution. Consider such an extreme solution x and, for the sake of contradiction, assume that it is non-integral.

We first show that for every agent $i\in\mathcal{A}$ and item $g\in\mathcal{G}$, at most one variable $x_{i,g,t}$ can be non-integral. Assume otherwise for agent $i\in\mathcal{A}$ and item $g\in\mathcal{G}$, and let t_1 and t_2 be the maximum and minimum elements in set $\{t : 0 < x_{i,g,t} < 1\}$. Let $\epsilon = \min\{1-x_{i,g,t_1}, x_{i,g,t_2}\}$ and consider the modified solution x' with $x'_{i,g,t_1} = x_{i,g,t_1}+\epsilon$ and $x'_{i,g,t_2} = x_{i,g,t_2}-\epsilon$, while x' has the same value with x on any triplet different than (i,g,t_1) and (i,g,t_2). Due to the feasibility of x, the new solution x' is clearly feasible. Furthermore, the objective value of x' is (at least) as high as that of x as it increases by $\epsilon\cdot v_i(g,t_1)$ and decreases by $\epsilon\cdot v_i(g,t_2)\leq \epsilon\cdot v_i(g,t_1)$. The last inequality follows since the valuations are monotone non-increasing. Hence, the solution x' has optimal objective value as well, and, furthermore, at least one additional integral variable compared to x: $x'_{i,g,t_1}=1$ if $1-x_{i,g,t_1}\leq x_{i,g,t_2}$ and $x'_{i,g,t_2}=0$ otherwise. Thus, solution x is not extreme, a contradiction.

Now, consider the bipartite graph $G=(\mathcal{A},\mathcal{G},E_x)$, where E_x contains the edge (i,g) if there exists t such that $x_{i,g,t}$ has non-integer value. Observe that G contains cycles. Indeed, if G was a tree, some node u in $\mathcal{A}\cup\mathcal{G}$ would have degree 1. If $u\in\mathcal{A}$, then $\sum_{g\in\mathcal{G},t\in[T]} x_{u,g,t}$ would include a single non-integer term (i.e., the weight $x_{u,g,t}$ of the single edge which is incident to node u). As T is integer, it would then be $\sum_{g\in\mathcal{G},t\in[T]} x_{u,g,t}\neq T$, violating the first LP constraint for agent u. If $u\in\mathcal{G}$, then $\sum_{i\in\mathcal{A},t\in[T]} x_{i,u,t}$ would include a single non-integer term. Again, this would imply that $\sum_{i\in\mathcal{A},t\in[T]} x_{i,u,t}\neq T$, violating the second LP constraint for item u.

Let C be a cycle in G. Since G is bipartite, C has even length and its edges can be partitioned into two matchings M_1 and M_2. For an edge (i,g) of E_x, let $t(i,g)$ be such that $x_{i,g,t(i,g)}$ is non-integer. Also, for a set of edges M of E_x, define $V(M)=\sum_{(i,g)\in M} x_{i,g,t(i,g)}\cdot v_i(g,t(i,g))$ and, without loss of generality, assume that $V(M_1)\geq V(M_2)$. Observe that $V(M_1)$ and $V(M_2)$ are simply the contribution to the objective value by the triplets $(i,g,t(i,g))$ corresponding to the edges (i,g) of M_1 and M_2, respectively. Now let

$$\epsilon = \min\left\{1-\max_{(i,g)\in M_1} x_{i,g,t(i,g)}, \min_{(i,g)\in M_2} x_{i,g,t(i,g)}\right\}$$

and modify the solution x to a new solution x' as follows:

- x' has the same value with x on any triplet that does not correspond to $(i,g,t(i,g))$ for an edge (i,g) of C.
- $x'_{i,g,t(i,g)} = x_{i,g,t(i,g)}+\epsilon$ for every $(i,g)\in M_1$, and
- $x'_{i,g,t(i,g)} = x_{i,g,t(i,g)}-\epsilon$ for every $(i,g)\in M_2$.

Clearly, the contribution of a triplet that does not correspond to triplet $(i,g,t(i,g))$ for an edge (i,g) of C to the objective value is the same under both solutions x and x'. The contribution from the triplets $(i,g,t(i,g))$ corresponding to edges (i,g) of M_1 increases by

$\epsilon \cdot V(M_1)$ in x' compared to x, and the contribution from the triplets $(i, g, t(i, g))$ corresponding to edges (i, g) of M_2 decreases by $\epsilon \cdot V(M_2) \leq \epsilon \cdot V(M_1)$. Hence, the objective value of solution x' is also optimal. Furthermore, solution x' has at least one additional integer variable compared to x: indeed, observe that $x'_{i_1,g_1,t(i,g)} = 1$ for some edge (i_1, g_1) of M_1 if $1 - \max_{(i,g)\in M_1} x_{i,g,t(i,g)} \leq \min_{(i,g)\in M_2} x_{i,g,t(i,g)}$ and $x'_{i_2,g_2,t(i_2,g_2)} = 0$ for some edge (i_2, g_2) of M_2, otherwise. Thus, solution x is not extreme, again a contradiction. □

References

[1] M. Aleksandrov, Envy-freeness up to one item: shall we add or remove resources?, CoRR, arXiv:2006.11312, 2020.

[2] G. Amanatidis, H. Aziz, G. Birmpas, A. Filos-Ratsikas, B. Li, H. Moulin, A.A. Voudouris, X. Wu, Fair division of indivisible goods: a survey, CoRR, arXiv:2208.08782, 2022.

[3] H. Aziz, I. Caragiannis, A. Igarashi, T. Walsh, Fair allocation of indivisible goods and chores, Auton. Agents Multi-Agent Syst. 36 (2022) 3.

[4] S. Barman, A. Biswas, S.K. Krishnamurthy, Y. Narahari, Groupwise maximin fair allocation of indivisible goods, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2017, pp. 9921–9922.

[5] S. Barman, G. Ghalme, S. Jain, P. Kulkarni, S. Narang, Fair division of indivisible goods among strategic agents, in: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2019, pp. 1811–1813.

[6] S. Barman, S.K. Krishnamurthy, R. Vaish, Finding fair and efficient allocations, in: Proceedings of the 19th ACM Conference on Economics and Computation (EC), 2018, pp. 557–574.

[7] S. Barman, S.K. Krishnamurthy, R. Vaish, Greedy algorithms for maximizing Nash social welfare, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2018, pp. 7–13.

[8] S. Barman, R.G. Sundaram, Uniform welfare guarantees under identical subadditive valuations, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020, pp. 46–52.

[9] K. Bérczi, E.R. Bérczi-Kovács, E. Boros, F.T. Gedefa, N. Kamiyama, T. Kavitha, Y. Kobayashi, K. Makino, Envy-free relaxations for goods, chores, and mixed items, CoRR, arXiv:2006.04428, 2020.

[10] U. Bhaskar, A.R. Sricharan, R. Vaish, On approximate envy-freeness for indivisible chores and mixed resources, in: Proceedings of the 25th International Conference on Randomization and Computation and the 24th International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX/RANDOM), 2021, pp. 1:1–1:23.

[11] A. Biswas, S. Barman, Fair division under cardinality constraints, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 91–97.

[12] E. Budish, The combinatorial assignment problem: approximate competitive equilibrium from equal incomes, J. Polit. Econ. 119 (2011) 1061–1103.

[13] I. Caragiannis, P. Kanellopoulos, M. Kyropoulou, On interim envy-free allocation lotteries, in: Proceedings of the 22nd ACM Conference on Economics and Computation (EC), 2021, pp. 264–284.

[14] I. Caragiannis, D. Kurokawa, H. Moulin, A.D. Procaccia, N. Shah, J. Wang, The unreasonable fairness of maximum Nash welfare, ACM Trans. Econ. Comput. 7 (2019) 1–32.

[15] P.H. Chan, X. Huang, Z. Liu, C. Zhang, S. Zhang, Assignment and pricing in roommate market, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI), 2016, pp. 446–452.

[16] X. Chen, Z. Liu, The fairness of leximin in allocation of indivisible chores, CoRR, arXiv:2005.04864, 2020.

[17] R. Cole, K. Ost, S. Schirra, Edge-coloring bipartite multigraphs in o (e logd) time, Combinatorica 21 (2001) 5–12.

[18] A. Farhadi, M. Hajiaghayi, M. Latifian, M. Seddighin, H. Yami, Almost envy-freeness, envy-rank, and Nash social welfare matchings, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI), 2021, pp. 5355–5362.

[19] D. Ferraioli, L. Gourvès, J. Monnot, On regular and approximately fair allocations of indivisible goods, in: 2014 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'14), 2014, pp. 997–1004.

[20] R. Freeman, E. Micha, N. Shah, Two-sided matching meets fair division, in: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), 2021, pp. 203–209.

[21] H.N. Gabow, R.E. Tarjan, Faster scaling algorithms for network problems, SIAM J. Comput. 18 (1989) 1013–1036.

[22] Y. Gafni, X. Huang, R. Lavi, I. Talgam-Cohen, Unified fair allocation of goods and chores via copies, CoRR, arXiv:2109.08671, 2021.

[23] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[24] S. Gollapudi, K. Kollias, B. Plaut, Almost envy-free repeated matching in two-sided markets, in: Proceedings of the 16th International Conference on Web and Internet Economics (WINE), 2020, pp. 3–16.

[25] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method, in: Geometric Algorithms and Combinatorial Optimization, Springer, Berlin, Heidelberg, 1988, pp. 64–101.

[26] H. Hosseini, K. Larson, R. Cohen, Matching with dynamic ordinal preferences, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI), 2015, pp. 936–943.

[27] B. Li, Y. Li, Fair resource sharing and dorm assignment, in: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2020, pp. 708–716.

[28] Y. Li, Y. Jiang, W. Wu, J. Jiang, H. Fan, Room allocation with capacity diversity and budget constraints, IEEE Access 7 (2019) 42968–42986.

[29] R.J. Lipton, E. Markakis, E. Mossel, A. Saberi, On approximately fair allocations of indivisible goods, in: Proceedings of the 5th ACM Conference on Electronic Commerce (EC), 2004, pp. 125–131.

[30] B. Plaut, T. Roughgarden, Almost envy-freeness with general valuations, SIAM J. Discrete Math. 34 (2020) 1039–1068.

[31] M.D. Plummer, L. Lovász, Matching Theory, Elsevier Science, 1986.

[32] A. Schrijver, Theory of Linear and Integer Programming, John Wiley & Sons, Inc., 1986.

[33] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory Comput. 3 (2007) 103–128.