Autonomous Agents and Multi-Agent Systems (2021) 35:15
https://doi.org/10.1007/s10458-021-09495-w

Probabilistic serial mechanism for multi-type resource allocation

Xiaoxi Guo[1] · Sujoy Sikdar[2] · Haibin Wang[1] · Lirong Xia[3] · Yongzhi Cao[1] · Hanpin Wang[1,4]

Accepted: 17 February 2021 / Published online: 3 April 2021

Abstract
In *multi-type resource allocation (MTRA) problems*, there are $d \geq 2$ *types* of items, and n agents who each demand one unit of items of each type and have *strict linear preferences* over *bundles* consisting of one item of each type. For MTRAs with indivisible items, our first result is an impossibility theorem that is in direct contrast to the single type ($d = 1$) setting: no mechanism, the output of which is always *decomposable* into a probability distribution over discrete assignments (where no item is split between agents), can satisfy both sd-efficiency and sd-envy-freeness. We show that this impossibility result is circumvented under the natural assumption of lexicographic preferences. We provide *lexicographic probabilistic serial (LexiPS)* as an extension of the *probabilistic serial (PS)* mechanism for MTRAs with lexicographic preferences, and prove that LexiPS satisfies sd-efficiency and sd-envy-freeness, retaining the desirable properties of PS. Moreover, LexiPS satisfies sd-weak-strategyproofness when agents are not allowed to misreport their importance orders. For MTRAs with divisible items, we show that the existing *multi-type probabilistic serial (MPS)* mechanism satisfies the stronger efficiency notion of lexi-efficiency, and is sd-envy-free under strict linear preferences and sd-weak-strategyproof under lexicographic preferences. We also prove that MPS can be characterized both by leximin-optimality and by item-wise ordinal fairness, and the family of eating algorithms which MPS belongs to can be characterized by lexi-efficiency.

Keywords Multi-type resource allocation · Probabilistic serial · LexiPS · MPS · Fractional assignment · sd-efficiency · sd-envy-freeness

1 Introduction

In this paper, we focus on extensions of the celebrated *probabilistic serial* (PS) mechanism [9] for the classical resource allocation problem [2, 9, 15, 33], to the *multi-type resource allocation problem* (MTRA) [32]. An MTRA involves n agents, $d \geq 2$ *types* of items which are not interchangeable, and one unit each of n items of each type. Each

✉ Yongzhi Cao
caoyz@pku.edu.cn

Extended author information available on the last page of the article

agent demands a *bundle* consisting of one item of each type and has *strict preferences* over all bundles. MTRAs may involve *divisible* items, like land and water resources [38], and computational resources such as CPU, memory, and storage in cloud computing [18–22]. Items may also be *indivisible*, where each item must be assigned fully to a single agent, like houses and cars [39, 40], and research papers and time slots in a seminar class [32].

Efficient and fair resource allocation for a single type of items ($d = 1$) has been well studied [1, 9, 34, 46]. Our work follows the line of research initiated by Bogomolnaia and Moulin [9], who proposed the probabilistic serial (PS) mechanism. The PS mechanism outputs a fractional assignment in multiple rounds by having all agents simultaneously "eat" shares of their favorite remaining items at a uniform and equal rate until some of the items are exhausted in each round. The remarkable properties of PS has encouraged several extensions: to the full preference domain, allowing indifferences [26, 28], to multi-unit demands [25], and to housing markets [3, 44].

PS is a popular prototype for mechanism designers due to the following reasons. (i) *Decomposability*: PS can be applied to allocating both divisible and indivisible items, since fractional assignments are always *decomposable* when $d = 1$, due to the Birkhoff-von Neumann theorem [6, 41]. In other words, a fractional assignment can be represented as a probability distribution over "discrete" assignments, where no item is split among agents. (ii) *Efficiency and fairness*: PS satisfies sd-efficiency and sd-envy-freeness which are desirable efficiency and fairness properties, respectively. They are based on the notion of *stochastic dominance* [9, 16]: given a strict preference relation over the items, an allocation p *weakly stochastically dominates* q, if at every item o, the total shares of item o and items strictly preferred to o in p, are at least the total shares of the same items in q.

Unfortunately, designing efficient and fair mechanisms for MTRAs with $d \geq 2$ types is more challenging, especially because direct applications of PS to MTRAs fail to simultaneously satisfy the two desirable properties of efficiency and fairness discussed above.

First, decomposability (property (i) above) does not always hold for fractional assignments in MTRAs as we show in the following example.

Example 1 Consider the MTRA with two agents, 1 and 2, two types of items, food (F) and beverages (B), and two items of each type $\{1_F, 2_F\}$ and $\{1_B, 2_B\}$, respectively. We show that the fractional assignment P below, where agent 1 gets a share of 0.5 units of $1_F 1_B$ and a share of of 0.5 units of $2_F 2_B$, is not decomposable.

Agent	P				Agent	P'			
	$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$		$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$
1	0.5	0	0	0.5	1	1	0	0	0
2	0	0.5	0.5	0	2	0	0	0	1

It is easy to see that the assignment P' above is the only assignment where $1_F 1_B$ is allocated fully to agent 1. Since agent 1 has a share of 0.5 units of $1_F 1_B$ in P, the probability for P' ought to be 0.5. Therefore, agent 2 should be allocated a share of 0.5 units of $2_F 2_B$ in P. However, agent 2 is not allocated $2_F 2_B$ in assignment P. Thus, P is not decomposable.

□

A natural idea is to decompose MTRA into d single-type instances, one for each type of items, and then apply PS or other mechanisms separately to each of them. Unfortunately, this does not work because it is unclear how to decompose agents' combinatorial preferences over bundles into separable preferences over items of the same type. More importantly, even when there is a natural way to do so, e.g. when agents' preferences are *lexicographic* and *separable*, meaning that every agent has an importance order over types to compare bundles and their preferences over a type do not depend on the items of other types, the following example shows that the fairness and efficiency properties (ii) above do not hold anymore.

Example 2 We continue to use the MTRA above and assume that agents' preferences over $\{1_F, 2_F\} \times \{1_B, 2_B\}$ are the following:

Agent	Preferences
1	$1_F 1_B \succ_1 1_F 2_B \succ_1 2_F 1_B \succ_1 2_F 2_B$
2	$1_F 1_B \succ_2 2_F 1_B \succ_2 1_F 2_B \succ_2 2_F 2_B$

We note that both agents prefer 1_F to 2_F, and 1_B to 2_B. Agent 1 considers F to be more important than B, while agent 2 considers B to be more important. In this way, we can decompose this MTRA into two single type resource allocation problems for F and B, respectively. It is easy to see that for each single type the only sd-efficient and sd-envy-free assignment is to give both agents 0.5 units of each item, yielding the decomposable fractional assignment Q by the mutual independence of each type. We show the assignments Q and Q' in the following:

Agent	Q				Agent	Q'			
	$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$		$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$
1	0.25	0.25	0.25	0.25	1	0.25	0.5	0	0.25
2	0.25	0.25	0.25	0.25	2	0.25	0	0.5	0.25

However, Q is inefficient as the decomposable assignment Q' stochastically dominates Q from both agents' perspectives. □

As we have observed, the two desirable properties of PS for single type resource allocation no longer obviously hold for MTRAs. Recently, Wang et al. [42] proposed *multi-type probabilistic serial (MPS)* mechanism as an extension of PS for MTRAs with *divisible* items, and proved that MPS is sd-efficient for general partial preferences, sd-envy-free for CP-net preferences [12], and sd-weak-strategyproof for CP-net preferences with a trivial dependency structure where all the types are independent. However, MPS does not satisfy decomposability and it is unclear whether similar extensions of the PS mechanism can be applied to the efficient and fair allocation of indivisible items because the outcome may not be decomposable. This leaves the following natural question:

How to design efficient and fair mechanisms for MTRAs with indivisible or divisible items?[1]

Our contributions For MTRAs with indivisible items, unfortunately, our impossibility theorem (Theorem 1) shows that no mechanism which satisfies sd-efficiency and sd-envy-freeness is guaranteed to always output decomposable assignments, if agents' preferences are allowed to be any strict linear orders over bundles. We also provide a tightened version of the impossibility result (Proposition 1). Fortunately, when agents' preferences are *lexicographic*, the impossibility theorem can be circumvented. To this end, we propose *lexicographic probabilistic serial mechanism (LexiPS)* and prove that it satisfies many of the desirable properties of PS: it is guaranteed to output a decomposable assignment, satisfy sd-efficiency and sd-envy-freeness (Theorem 2), and satisfy sd-weak-strategyproofness when agents do not lie about their importance orders over types (Theorem 3).

For MTRAs with divisible items, we show that when agents' preferences are linear orders over all bundles of items, the MPS mechanism proposed by Wang et al. [42] satisfies lexi-efficiency (Theorem 4) which is a stronger notion of efficiency than sd-efficiency. Indeed, we show that lexi-efficiency is characterized by the *no-generalized-cycle* condition, which is a sufficient condition for sd-efficiency but not a necessary one (Proposition 2). We also prove that *every* lexi-efficient assignment can be computed by some algorithm in the family of *eating* algorithms (Theorem 5), of which MPS is a member. Importantly, MPS retains sd-envy-freeness (Proposition 3), and when agents' preferences are further assumed to be lexicographic, MPS satisfies sd-weak-strategyproofness (Theorem 6). Finally, we characterize MPS both by leximin-optimality and by item-wise ordinal fairness (Theorem 7). However, the output of MPS is not always decomposable (Remark 4) even under lexicographic preferences, making it unsuitable for MTRAs with indivisible items.

Related work and discussions To the best of our knowledge, our paper provides the first results on designing efficient and fair mechanisms based on stochastic dominance for MTRAs *with indivisible items*. Budish et al. [13] considered the multi-unit demand resource allocation problem with constraints and provided a condition which guarantees that a constraint structure is universally implementable, but this result does not apply to MTRAs because their setting is different from ours as discussed below. Wang et al. [42] considered MTRAs and Chatterji and Liu [14] and Nguyen et al. [35] considered a related problem of assigning bundles of multiple items of a single type, but they did not provide a decomposable mechanism that is both sd-efficient and sd-envy-free. Despite our impossibility result for general MTRAs (Theorem 1), the LexiPS mechanism we provide and its properties allows us to deliver the following positive message: *it is possible to design efficient and fair mechanisms for indivisible items under the natural domain restriction of lexicographic preferences*.

MTRAs were introduced by Moulin [33], and were more recently explicitly formulated in the form presented in the paper by Mackin and Xia [32], who provided a characterization of serial dictatorships satisfying strategyproofness, neutrality, and non-bossiness for MTRAs. In a similar vein, Sikdar et al. [39, 40] considered multi-type housing markets [33].

Wang et al. [42] studied fractional mechanisms for MTRAs when agents' preferences may be partial orders. In that paper, MPS, MRP, and MGD are proposed as extensions of the famous probabilistic serial (PS) [9] and random priority (RP) [1] mechanisms for

[1] Note that for indivisible items, the (fractional) output of a mechanism must be decomposable.

allocating items of a single type to MTRAs. Our results which describe the properties of MPS for MTRAs with strict linear preferences complement the results in [42] for partial preferences in the following aspects:

(i) MPS satisfies sd-efficiency for the unrestricted domain of partial orders [42], while we prove that MPS satisfies lexi-efficiency, which is a stronger notion of efficiency than sd-efficiency, for the unrestricted domain of linear orders.
(ii) MPS satisfies sd-weak-strategyproofness when agents' preferences are CP-nets with a trivial dependency structure such that all the types are independent [42], while we prove the result when agents' preferences are lexicographic with possibly different importance orders on types.
(iii) We provide characterizations of MPS, which are not considered in [42].

Moreover, we show that no mechanism satisfies sd-efficiency, sd-envy-freeness, and decomposability simultaneously, even under strict linear preferences. Therefore, MPS, MRP, and MGD [42] also fail to satisfy all the three properties simultaneously. We prove that this impossibility result can be circumvented under the natural restriction of lexicographic preferences by providing LexiPS as a mechanism for MTRAs with lexicographic preferences that satisfies all the three properties. To the best of our knowledge, the design of mechanisms satisfying all the three properties of efficiency, fairness, and decomposability for MTRAs is not considered in [42] or other previous works.

Chatterji and Liu [14] and Nguyen et al. [35] considered a problem closely related to the MTRA. In their setting, every item must be allocated to some agent, and agents may be allocated bundles consisting of multiple items and have strict preferences over bundles where the empty allocation may be preferred to some subset of possible bundles. We refer to the problem in their setting as the bundle assignment problem. Due to the fact that in MTRAs, agents have strict linear preferences over bundles consisting of one item of each type and prefer the empty allocation to all the other bundles, the MTRA is a special case of the bundle assignment problem. Chatterji and Liu [14] and Nguyen et al. [35] proposed the probabilistic serial rule for bundles (PSB) and bundled probabilistic serial (BPS) mechanisms respectively for the bundle assignment problem. When applied to MTRAs, MPS [42] is similar to PSB and BPS because all the three mechanisms have agents consume their current most preferred bundles till the bundles are unavailable before turning to other bundles. Although MTRA is a special case of the bundle assignment problem, and the MPS mechanism is similar to the PSB and BPS mechanisms, our results are complementary and not directly comparable to the results in these previous works:

(i) Chatterji and Liu [14] and Nguyen et al. [35] proved that PSB and BPS satisfy sd-efficiency for the bundle assignment problem, respectively, whereas we prove that MPS satisfies lexi-efficiency for MTRAs which is a stronger notion of efficiency than sd-efficiency.
(ii) Technically, the stronger efficiency guarantee of MPS is due to our characterization of lexi-efficiency by the no-generalized-cycle condition which is similar to the strong unbalancedness condition [14], but strong unbalancedness does not have a similar characterization.
(iii) Chatterji and Liu [14] considered the domain restriction of *essentially mononotonic preferences*, which is incomparable to lexicographic preferences considered in our

work. Therefore, their results for the essentially monotonic domain of preferences do not directly apply to our setting.

(iv) Additionally, while PSB is not sd-weak-strategyproof for the bundle assignment problem [14], we prove that MPS is sd-weak-strategyproof for MTRAs when restricted to lexicographic preferences.

(v) Nguyen et al. [35] considered the implementation of the output of BPS for indivisible items in the relaxed economy where there are extra copies of items with free disposal. This is not comparable with decomposability of an assignment in our setting, where we do not have either free disposal or extra copies of items. Further, we prove that for general MTRAs, it is impossible to design an sd-efficient and sd-envy-free mechanism whose output is guaranteed to be decomposable.

Another related problem setting considered in Aziz and Kasajima [4], Kojima [29], and Budish et al. [13], is the one where items are of a single type and agents may demand multiple units of items. We call it the multi-unit demand resource allocation problem. These works considered fractional assignments that consist of shares of items, which are fundamentally different from our work where we consider fractional assignments composed of shares of bundles for MTRAs. Specifically, a fractional assignment on items may imply different fractional assignments on bundles, each with possibly different properties. Importantly, the extension of the notion of stochastic dominance in terms of shares of items in these works is also fundamentally different from the notion of stochastic dominance for bundles in our paper, and they are not comparable. Therefore, their results on notions of efficiency and fairness based on the stochastic dominance do not apply to our setting. Kojima [29] provided an extension of PS which is sd-efficient and sd-envy-free but not sd-weak-strategyproof, and an impossibility result that no mechanism can satisfy these three properties simultaneously in the multi-unit demand resource allocation problem. Aziz and Kasajima [4] provided impossibility results involving sd-efficiency and sd-weak-strategyproofness for the problem. Budish et al. [13] considered the multi-unit demand resource allocation problem with constraints and provide two mechanisms, including an extension of PS named generalized probabilistic serial (GPS) which generalizes the one in Kojima [29]. The MTRA with divisible items may also be viewed as a version of the cake-cutting problem with multiple cakes [18, 27, 31, 36] and agents having ordinal preferences over combinations of pieces from each cake.

The lexicographic preference is a natural restriction on preference domain in resource allocation [19, 39, 40] and combinatorial voting [11, 30, 43]. Saban and Sethuraman [37] showed that PS is efficient, envy-free, and strategy-proof under lexicographic preferences on allocations. Fujita et al. [19] considered the allocation problem which allows agents to receive multiple items and agents rank the groups of items lexicographically. Our work follows in this research agenda of natural domain restrictions on agents' preferences to circumvent impossibility results in guaranteeing efficiency and fairness.

Structure of the paper The rest of the paper is organized as follows. In Sect. 2, we define the MTRA problem, and provide definitions of desirable efficiency and fairness properties. Section 3 is the impossibility result for MTRAs with indivisible items. In Sect. 4, we propose LexiPS for MTRAs with indivisible items under lexicographic preferences, which satisfies sd-efficiency and sd-envy-freeness, and it is sd-weak-strategyproof when agents do not lie about importance orders. In Sect. 5, we show the properties of MPS for MTRAs with divisible items under strict linear preferences and provide two characterizations for MPS. In Sect. 6, we summarize the contributions of our paper and discuss directions for future work.

2 Preliminaries

An MTRA is given by a tuple (N, M) with a preference profile R. Let $N = \{1, \dots, n\}$ be the set of agents and $M = D_1 \cup \cdots \cup D_d$ be the set of all the items where D_i is the set of n items of type i for each $i \leq d$. For all $h \neq i$, we have $D_i \cap D_h = \emptyset$. There is one unit of *supply* of each item in M. We use $\mathcal{D} = D_1 \times \cdots \times D_d$ to denote the set of *bundles*. Each bundle $\mathbf{x} \in \mathcal{D}$ is a d-vector and each component refers to an item of each type. We use $o \in \mathbf{x}$ to indicate that bundle $\mathbf{x}$ contains item o. In an MTRA, each agent demands one unit of item of each type.

A *preference profile* is denoted by $R = (\succ_j)_{j \leq n}$, where $\succ_j$ represents agent j's preference as a *strict linear preference*, i.e. the strict linear order over $\mathcal{D}$. Let $\mathcal{R}$ be the set of all the preference profiles.

A fractional *allocation* is a $|\mathcal{D}|$-vector, describing the fractional share of each bundle allocated to an agent. Let Π be the set of all the possible fractional allocations. For any $p \in \Pi$, $\mathbf{x} \in \mathcal{D}$, we use $p_{\mathbf{x}}$ to denote the share of $\mathbf{x}$ assigned by p. A fractional *assignment* is a $n \times |\mathcal{D}|$-matrix $P = [p_{j,\mathbf{x}}]_{j \leq n, \mathbf{x} \in \mathcal{D}}$, where (i) $p_{j,\mathbf{x}} \in [0, 1]$ is the fractional share of $\mathbf{x}$ allocated to agent j for each $j \leq n, \mathbf{x} \in \mathcal{D}$, (ii) $\sum_{\mathbf{x} \in \mathcal{D}} p_{j,\mathbf{x}} = 1$, fulfilling the demand of each agent $j \leq n$, (iii) $\sum_{j \leq n, \mathbf{x} \in Z_o} p_{j,\mathbf{x}} = 1$, respecting the unit supply of each $o \in M$ and $Z_o = \{\mathbf{x} \in \mathcal{D} | o \in \mathbf{x}\}$. For each $j \leq n$, the j-th row of P, denoted by P_j, represents agent j's fractional allocation in P. We use $\mathcal{P}$ to denote the set of all possible fractional assignments. A *discrete assignment* A, is an assignment where each agent is assigned a share of one unit of a bundle, and each item is fully allocated to some agent[2]. It follows that a discrete assignment is represented by a matrix where each element is either 0 or 1. We use $\mathcal{A}$ to denote the set of all the discrete assignment matrices.

A *mechanism* f is a mapping from preference profiles to fractional assignments. For any profile $R \in \mathcal{R}$, we use $f(R)$ to refer to the fractional assignment output by f and $f(R)_j$ refer to agent j's fractional allocation in $f(R)$ for any agent $j \leq n$ accordingly.

2.1 Desirable properties

We use the notion of stochastic dominance to compare fractional assignments and recall the desirable notions of efficiency and fairness in [42] for MTRAs.

Definition 1 *(stochastic dominance* [42]*)* Given a preference relation $\succ$ over $\mathcal{D}$, the *stochastic dominance* relation associated with $\succ$, denoted by $\succeq^{sd}$, is a partial ordering over Π such that for any pair of fractional allocations $p, q \in \Pi$, p (weakly) *stochastically dominates* q, denoted by $p \succeq^{sd} q$, if for any $\mathbf{y} \in \mathcal{D}$, $\sum_{\mathbf{x} \in U(\succ, \mathbf{y})} p_{\mathbf{x}} \geq \sum_{\mathbf{x} \in U(\succ, \mathbf{y})} q_{\mathbf{x}}$, where $U(\succ, \mathbf{y}) = \{\mathbf{x} \in \mathcal{D} | \mathbf{x} \succ \mathbf{y}\} \cup \{\mathbf{y}\}$.

2 For for indivisible items, discrete assignments refer to deterministic assignments in the papers about randomization.

We also define $\nsucceq^{sd}$: $p \nsucceq^{sd} q$ if $p \succeq^{sd} q$ is not true. The stochastic dominance order can also be extended to fractional assignments. For $P, Q \in \mathscr{P}$ and $j \le n$, we assume that agent j only cares about her own allocations P_j and Q_j. If $P_j \succeq_j^{sd} Q_j$, agent j weakly prefers P to Q, i.e. $P \succeq_j^{sd} Q$. Therefore, we say that P weakly stochastically dominates Q, denoted by $P \succeq^{sd} Q$, if $P \succeq_j^{sd} Q$ for any $j \le n$. We also extend $\nsucceq^{sd}$ to assignments: $P \nsucceq^{sd} Q$ if $P \succeq^{sd} Q$ is not true. It is easy to prove that $P \succeq_j^{sd} Q$ and $Q \succeq_j^{sd} P$ if and only if $P_j = Q_j$.

Definition 2 *(*sd-efficiency [42]*)* Given an MTRA (N, M) and a preference profile R, a fractional assignment P is sd-efficient if there is no other fractional assignment $Q \neq P$ such that $Q \succeq_j^{sd} P$ for any $j \le n$. Correspondingly, if for any $R \in \mathscr{R}$, $f(R)$ is sd-efficient, then we say that mechanism f satisfies sd-efficiency.

Definition 3 *(*sd-envy-freeness [42]*)* Given an MTRA (N, M) and a preference profile R, a fractional assignment P is sd-envy-free if $P_j \succeq_j^{sd} P_k$ for any two agents $j, k \le n$. Correspondingly, if for any $R \in \mathscr{R}$, $f(R)$ is sd-envy-free, then we say that mechanism f satisfies sd-envy-freeness.

Definition 4 *(*sd-weak-strategyproofness [42]*)* Given an MTRA (N, M) and a preference profile R, a mechanism f satisfies sd-weak-strategyproofness if for any profile $R \in \mathscr{R}$ and agent $j \le n$, it holds that

$$f(R') \succeq_j^{sd} f(R) \implies f(R')_j = f(R)_j$$

for any $R' \in \mathscr{R}$ where $R' = (\succ_j', \succ_{-j})$ and $\succ_{-j}$ denotes the preferences of agents in the set $N \setminus \{j\}$.

Besides stochastic dominance, we introduce the *lexicographic dominance* relation [37] to compare pairs of fractional allocations, by comparing the components of their respective vector representations one by one according to the agent's preference.

Definition 5 *(lexicographic dominance)* Given a preference relation $\succ$ and a pair of allocations p and q, the *lexicographic dominance* relation associated with $\succ$, denoted by $\succ^{lexi}$, is a strict ordering over Π such that p lexicographically dominates q, denoted by $p \succ^{lexi} q$, if there exist a bundle $\mathbf{y}$ such that $p_{\mathbf{y}} > q_{\mathbf{y}}$, and for any $\mathbf{x} \succ \mathbf{y}$, $p_{\mathbf{x}} \ge q_{\mathbf{x}}$.

Given assignments P and Q, we say $Q \succ^{lexi} P$ if there exists a set of agents $N' \subseteq N$ and $N' \neq \emptyset$ such that $Q_k \succ_k^{lexi} P_k$ ($Q \succ_k^{lexi} P$ for short) for any agent $k \in N'$ and $Q_j = P_j$ for any agent $j \in N \setminus N'$. Note that for two different assignments, stochastic dominance implies lexicographic dominance, but the converse does not hold.

Definition 6 *(*lexi-efficiency*)* Given a preference profile R, the fractional assignment P is lexi-efficient if there is no $Q \in \mathscr{P}$ such that $Q \succ^{lexi} P$. A fractional assignment algorithm f satisfies lexi-efficiency if $f(R)$ is lexi-efficient for any $R \in \mathscr{R}$.

Example 3 To compare lexicographic dominance with stochastic dominance, we revisit the MTRA in Example 2 and consider the relation of the following assignment Q and Q'':

Agent	Q				Agent	Q''			
	1_F1_B	1_F2_B	2_F1_B	2_F2_B		1_F1_B	1_F2_B	2_F1_B	2_F2_B
1	0.25	0.25	0.25	0.25	1	0.5	0	0	0.5
2	0.25	0.25	0.25	0.25	2	0.5	0	0	0.5

According to R in Example 2, we have that Q'' does not stochastically dominate Q, because $\sum_{\mathbf{x}\in U(\succ_1,2_F1_B)} q_{1,\mathbf{x}} = 0.75 > 0.5 = \sum_{\mathbf{x}\in U(\succ_1,2_F1_B)} q''_{1,\mathbf{x}}$. However, we have $Q'' \succ^{lexi} Q$ because $q''_{1,1_F1_B} = 0.5 > 0.25 = q_{1,1_F1_B}$ and $q''_{2,1_F1_B} = 0.5 > 0.25 = q_{2,1_F1_B}$, where 1_F1_B is the most preferred bundle for both agents. □

3 Efficiency and fairness for MTRAs with indivisible items

In this section, we show an impossibility result in Theorem 1 that no mechanism satisfying sd-envy-freeness and sd-efficiency is guaranteed to output decomposable assignments. This is unlike the case of resource allocation problems with a single type of items, where by sd-envy-freeness of PS, every fractional assignment is decomposable, i.e. every fractional assignment P can be decomposed into a probability distribution over the set of discrete assignments $\mathscr{A}$ as follows:

$$P = \sum_{A\in\mathscr{A}} \alpha^A \cdot A.$$

Here, each A is a discrete assignment that assigns each item wholly to some agent. We note that $\sum_{A\in\mathscr{A}} \alpha^A = 1$. It follows that such a decomposable assignment can be applied to the problem of allocating indivisible items as a lottery over $\mathscr{A}$ where a discrete assignment A is selected with probability α^A. This result does not necessarily hold in MTRAs, which leads to the impossibility result.

Theorem 1 *For any MTRAs with $d \geq 2$ where agents are allowed to submit any strict linear orders over bundles, no mechanism that satisfies sd-efficiency and sd-envy-freeness always outputs decomposable assignments.*

Proof Suppose for the sake of contradiction that there exists a mechanism f satisfying sd-efficiency and sd-envy-freeness and $f(R)$ is always decomposable for any $R \in \mathscr{R}$. We first provide a proof for MTRAs where there are $d = 2$ types and $|N| = 2$ agents, and then extend it to the general case. Let R be the following preference profile and $Q = f(R)$.

Agent	Preferences
1	$1_F1_B \succ_1 1_F2_B \succ_1 2_F2_B \succ_1 2_F1_B$
2	$1_F2_B \succ_2 2_F1_B \succ_2 1_F1_B \succ_2 2_F2_B$

We show that if Q is sd-envy-free and decomposable, it fails to satisfy sd-efficiency. There are only four discrete assignments which assign $1_F1_B, 1_F2_B, 2_F1_B, 2_F2_B$ to agent 1, respectively. Since Q is decomposable, it can be represented as the following assignment.

We also provide an assignment P which is not decomposable since it does not satisfy the constraints for Q.

Agent	P				Agent	Q			
	1_F1_B	1_F2_B	2_F1_B	2_F2_B		1_F1_B	1_F2_B	2_F1_B	2_F2_B
1	0.5	0	0	0.5	1	v	w	y	z
2	0	0.5	0.5	0	2	z	y	w	v

Here v, w, y and z are probabilities of these four discrete assignments which satisfy $v+w+y+z=1$. We trivially have that $\sum_{\mathbf{x}\in U(\succ_1,2_F1_B)} q_{1,\mathbf{x}} = 1 = \sum_{\mathbf{x}\in U(\succ_1,2_F1_B)} q_{2,\mathbf{x}}$. In addition, we have the following inequalities by sd-envy-freeness in terms of agent 1:

$$\begin{aligned}
\sum_{\mathbf{x}\in U(\succ_1,1_F1_B)} q_{1,\mathbf{x}} &= v \geq z = \sum_{\mathbf{x}\in U(\succ_1,1_F1_B)} q_{2,\mathbf{x}}\\
\sum_{\mathbf{x}\in U(\succ_1,1_F2_B)} q_{1,\mathbf{x}} &= v+w \geq z+y = \sum_{\mathbf{x}\in U(\succ_1,1_F2_B)} q_{2,\mathbf{x}}\\
\sum_{\mathbf{x}\in U(\succ_1,2_F2_B)} q_{1,\mathbf{x}} &= v+w+z \geq z+y+v = \sum_{\mathbf{x}\in U(\succ_1,2_F2_B)} q_{2,\mathbf{x}}
\end{aligned}$$

Similarly, we have that $y \geq w$ and $y+w+z \geq w+y+v$ for agent 2. Thus $w=y, v=z$ and $v+w=y+z=0.5$. Because Q is sd-efficient, $P \not\succeq^{sd} Q$. Suppose that $P \not\succeq_1^{sd} Q$. Therefore, at least one of the following inequalities is true:

$$\begin{aligned}
\sum_{\mathbf{x}\in U(\succ_1,1_F1_B)} q_{1,\mathbf{x}} &= v > 0.5 = \sum_{\mathbf{x}\in U(\succ_1,1_F1_B)} p_{1,\mathbf{x}}\\
\sum_{\mathbf{x}\in U(\succ_1,1_F2_B)} q_{1,\mathbf{x}} &= v+w > 0.5 = \sum_{\mathbf{x}\in U(\succ_1,1_F2_B)} p_{1,\mathbf{x}}\\
\sum_{\mathbf{x}\in U(\succ_1,2_F2_B)} q_{1,\mathbf{x}} &= v+w+z > 1 = \sum_{\mathbf{x}\in U(\succ_1,2_F2_B)} p_{1,\mathbf{x}}\\
\sum_{\mathbf{x}\in U(\succ_1,2_F1_B)} q_{1,\mathbf{x}} &= v+w+z+y > 1 = \sum_{\mathbf{x}\in U(\succ_1,2_F1_B)} p_{1,\mathbf{x}}
\end{aligned} \tag{1}$$

Since $v=z \leq v+w=y+z=0.5$, none of the inequalities in (1) hold, which means that $P \succeq_1^{sd} Q$. With a similar analysis, we can also obtain that $P \succeq_2^{sd} Q$. Together we have the fact that $P \succeq^{sd} Q$ and $P \neq Q$, which is contradictory to the assumption.

Now, we prove the theorem for the general case of MTRAs where $d \geq 2$ and $|N| \geq 2$ by constructing a profile R' for arbitrary numbers of types d and agents n, by extending the profile R for the case of $d=2$ and $|N|=2$ we constructed above. We use i which ranges from 1 to d to refer to the types in the problem. W.l.o.g. we use F and B to denote types 1 and 2, respectively. Let o_i be an arbitrary item of type i, and $o_{\{i_1,i_2,\dots\}}$ be the partial bundle containing o_{i_1}, o_{i_2}, and other o_h with h in brackets. For convenience, we define that $o_{[h,i]}$ refers to the partial bundle which contains $o_h, o_{h+1}, \dots, o_i$ for some type $h \leq i$.

In R', agent k prefers the bundle $k_{[1,d]}$ to all the other bundles for any $k \geq 3$. For agents 1 and 2, their preferences in R' are as follows:

Agent	Preferences
1	$1_F 1_B 1_{[3,d]} \succ_1 1_F 2_B 2_{[3,d]} \succ_1 2_F 2_B 2_{[3,d]} \succ_1 2_F 1_B 1_{[3,d]} \succ_1$ others
2	$1_F 2_B 2_{[3,d]} \succ_2 2_F 1_B 1_{[3,d]} \succ_2 1_F 1_B 1_{[3,d]} \succ_2 2_F 2_B 2_{[3,d]} \succ_2$ others

It follows from our construction that in any sd-efficient and sd-envy-free assignment, agent k always gets the bundle $k_{[1,d]}$ for any $k \geq 3$.

Now, let us consider the assignment $Q' = f(R')$. It suffices to consider the partial assignment $Q'_{\{1,2\}}$ which only contains the allocations of agents 1 and 2 in the rest of the proof:

Agent	$Q'_{\{1,2\}}$				
	$1_F 1_B 1_{[3,d]}$	$1_F 2_B 2_{[3,d]}$	$2_F 1_B 1_{[3,d]}$	$2_F 2_B 2_{[3,d]}$	others
1	v	w	y	z	0
2	z	y	w	v	0

It is easy to see that agents 1 and 2 can only get shares of the bundles containing 1_i or 2_i for type i. Moreover, we claim that, in the assignment Q', both agents do not have shares of any bundles that are not in the set $\mathscr{D} = \{1_F 1_B 1_{[3,d]}, 1_F 2_B 2_{[3,d]}, 2_F 2_B 2_{[3,d]}, 2_F 1_B 1_{[3,d]}\}$ because Q' is sd-efficient. Suppose w.l.o.g. that agent 1 obtains s units of some bundle $\mathbf{x} \notin \mathscr{D}$ in Q'. Observe that $\mathbf{x}$ must contain one of the following partial bundles: $1_F 1_B, 1_F 2_B, 2_F 1_B$ or $2_F 2_B$.

We first consider the case the $\mathbf{x}$ contains $1_F 1_B$ but $\mathbf{x} \neq 1_F 1_B 1_{[3,d]}$. It means that agent 1 obtains s units of $\mathbf{x} = 1_F 1_B 1_{\{h_1,h_2,\ldots,h_{|H|}\}} 2_{\{i_1,i_2,\ldots,i_{|I|}\}}$ for $H = \{h_1, h_2, \ldots\}$ and $I = \{i_1, i_2, \ldots\}$ with $H \cap I = \emptyset$, $H \cup I = \{3, 4, \ldots, d\}$, and $I \neq \emptyset$. Then, by decomposability of Q', we can infer that agent 2 obtains s units of the bundle $\mathbf{y} = 2_F 2_B 2_{\{h_1,h_2,\ldots,h_{|H|}\}} 1_{\{i_1,i_2,\ldots,i_{|I|}\}}$. However, agents 1 and 2 can obtain preferable allocations if they trade their shares of some partial bundles in $\mathbf{x}$ and $\mathbf{y}$. One specific way to achieve this is that agent 1 trades s units of $2_{\{i_1,i_2,\ldots,i_{|I|}\}}$ in $\mathbf{x}$ with agent 2 for $1_{\{i_1,i_2,\ldots,i_{|I|}\}}$ in $\mathbf{y}$. In this way, agents 1 gets s units of $1_F 1_B 1_{[3,d]}$ instead of $\mathbf{x}$ and 2 get s units of $2_F 2_B 2_{[3,d]}$ instead of $\mathbf{y}$, and from their preferences in R', we know $1_F 1_B 1_{[3,d]} \succ_1 \mathbf{x}$ and $2_F 2_B 2_{[3,d]} \succ_2 \mathbf{y}$. This is a contradiction to our assumption that f and therefore $Q' = f(R')$ is sd-efficient.

It is easy to see that this argument can be extended to the other cases when $\mathbf{x}$ contains $1_F 2_B, 2_F 1_B$ or $2_F 2_B$. This proves that the partial assignment $Q'_{\{1,2\}}$ of the assignments to agents 1 and 2 only involves positive shares of the four bundles in $\mathscr{D} = \{1_F 1_B 1_{[3,d]}, 1_F 2_B 2_{[3,d]}, 2_F 2_B 2_{[3,d]}, 2_F 1_B 1_{[3,d]}\}$ and 0 share of any bundles outside the set $\mathscr{D}$.

Then, by a similar argument to the case with $d = 2$ types and $n = 2$ agents above, we have a contradiction to our assumption that $Q'_{\{1,2\}}$ is sd-efficient, sd-envy-free, and decomposable simultaneously, which also means that f fails to satisfy all the three properties.

□

In Proposition 1 below, we provide a tighter version of the impossibility result in Theorem 1, by showing that even under LP-tree preferences [11] which is a restriction on the domain of strict linear preferences, no mechanism that is guaranteed to output decomposable assignments can simultaneously satisfy sd-weak-efficiency [8, 23] and sd-weak-envy-freeness [9] which are weaker notions of efficiency and fairness than sd-efficiency and sd-envy-freeness, respectively. We define each of these notions formally below before stating Proposition 1. We use $D_i(\mathbf{x})$ to denotes the item of type i in bundle $\mathbf{x}$.

LP-tree preference: a strict preference relation $\succ$ over $\mathscr{D} = D_1 \times \cdots \times D_d$ is an LP-tree preference if there exists a rooted directed tree (V, E) where (i) the node $v \in V$ is labeled by a type i with a strict linear order over D_i attached to it, (ii) each type occurs only once on each branch, and (iii) each outgoing edge from v is labeled by an item of type i, such that for any two bundles $\mathbf{x}, \mathbf{y} \in \mathscr{D}$, $\mathbf{x} \succ \mathbf{y}$ if there exists a node v with type i which satisfies that $o_g = D_g(\mathbf{x}) = D_g(\mathbf{y})$ occurs on the path from root to node v for any type g labeling an ancestor of node v in the tree and $D_i(\mathbf{x}) \succ^i D_i(\mathbf{y})$ where $\succ^i$ is the strict linear order over D_i attached to node v.

sd-weak-efficiency: a mechanism f satisfies sd-weak-efficiency if for any profile $R \in \mathscr{R}$, there is no fractional assignment P such that for any agent $j \le n$, it holds that $P \succeq_j^{sd} f(R)$ and $|\{k \le n | P_k \neq f(R)_k\}| = 2$.

sd-weak-envy-freeness: a mechanism f satisfies sd-weak-envy-freeness if for any profile $R \in \mathscr{R}$, and for any two agents $j, k \le n$, it holds that $f(R)_k = f(R)_j$ whenever $f(R)_k \succeq_j^{sd} f(R)_j$.

Proposition 1 *For MTRAs with LP-tree preferences, no mechanism that satisfies sd-weak-efficiency and sd-weak-envy-freeness always outputs decomposable assignments.*

The full proof of Proposition 1 is provided in "Proof of Proposition 1" in Appendix.

Remark 1 We note that all the three properties in Theorem 1, sd-efficiency, sd-envy-freeness, and decomposability, are necessary, i.e. there exist mechanisms satisfying any two of them. We show the example for each combination below:

(i) sd-efficiency and sd-envy-freeness: Wang et al. [42] and Section 5 of the paper show that MPS is sd-efficient and sd-envy-free.
(ii) sd-efficiency and decomposability: The extension of serial dictatorship [32] for MTRAs outputs a Pareto-optimal discrete assignment for any preference profile. Such an assignment is sd-efficient and trivially decomposable.
(iii) sd-envy-freeness and decomposability: Let f be the mechanism which outputs the same assignment P where every agent gets $1/|\mathscr{D}|$ units of each bundle for any preference profile. It is easy to check that P can be represented as a uniform distribution over all the possible discrete assignments. □

4 MTRAs with indivisible items and lexicographic preferences

Faced with the impossibility results of Theorem 1 and Proposition 1, a natural question to ask is whether it can be circumvented under a reasonable restriction on the problem domain. In this section, we show that the natural domain restriction of lexicographic preferences provides one such avenue. We develop LexiPS as a specialized mechanism for

MTRAs when agents' preferences are lexicographic, and prove that LexiPS retains the desirable properties of PS, namely sd-efficiency and sd-envy-freeness, and is guaranteed to output decomposable assignments meaning that it can be applied to MTRAs with indivisible items. An agent with a lexicographic preference over $\mathcal{D}$ has an *importance order* over the types and preferences over items of each type, and she compares two bundles by comparing the items of each type in the two bundles one by one according to her importance order on types, and prefers the bundle with the preferable item of the most important type at which the two bundles have different items. We define the lexicographic preference relation formally below. Before we begin, we note that we use the following notation throughout: $D_i(\mathbf{x})$ refers to the item of type i in the bundle $\mathbf{x}$ for any $\mathbf{x} \in \mathcal{D}$ and $i \leq d$.

Definition 7 *(lexicographic preference relation)* A strict preference relation $\succ$ over $\mathcal{D} = D_1 \times \cdots \times D_d$ is *lexicographic* if there exist (i) an *importance order*, i.e. a strict linear order $\rhd$ over types $\{1, \ldots, d\}$ and (ii) for each type $i \leq d$, a strict linear order $\succ^i$ over D_i such that for any two bundles $\mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\mathbf{x} \succ \mathbf{y}$ if there exists a type i satisfying $D_i(\mathbf{x}) \succ^i D_i(\mathbf{y})$ and $D_h(\mathbf{x}) = D_h(\mathbf{y})$ for any $h \rhd i$.

We note that although lexicographic preference relation and lexicographic dominance look similar, a lexicographic preference relation is used to compare bundles and represent agents' preferences, while lexicographic dominance is used to compare allocations or assignments consisting of shares of bundles. For any agent $j \leq n$, her preference $\succ_j$ is *lexicographic* if there exists an importance order $\rhd_j$ and strict linear orders $\succ_j^i$ similar to Definition 7. For example, the preference $1_F 2_B \succ 1_F 1_B \succ 2_F 2_B \succ 2_F 1_B$ is lexicographic with the importance order $F \rhd B$ and strict linear orders $1_F \succ^F 2_F$ and $2_B \succ^B 1_B$ over the types F and B, respectively. If every agent has a lexicographic preference in an MTRA, then we say that it is an MTRA with lexicographic preferences.

4.1 The LexiPS mechanism

Before going any further with LexiPS, we introduce some notations for ease of exposition. We use P^i to denote the fractional assignment of items of type i w.r.t. P. The assignment P^i is a $|N| \times |D_i|$ matrix with $p^i_{j,o} = \sum_{o \in \mathbf{x}, \mathbf{x} \in \mathcal{D}} p_{i,\mathbf{x}}$ representing the total shares of bundles containing items o of type i and consumed by agent j. To distinguish from single type fractional assignments, we refer to the fractional assignments for MTRAs as multi-type fractional assignments. Besides, for $o \in D_i$, we overload the notation of the upper contour set $U(\succ^i, o)$ to refer to the items of type i that are either strictly preferred or equal to o w.r.t. $\succ^i$.

Algorithm 1 LexiPS

Input: An MTRA (N,M), a lexicographic preference profile R.
For each $o \in M$, $supply(o) \leftarrow 1$. For each $i \leq d$, $P^i \leftarrow 0^{n\times n}$. $P \leftarrow 0^{n\times|\mathscr{D}|}$.
loop for d **phases**
  Identify top type i_j for each agent $j \leq n$.
  //For the kth phase, i_j is the kth most important type w.r.t. $\rhd_j$.
  for $i \leq d$ **do**
    $t \leftarrow 0$.
    $N^i = \{j \leq n | i_j = i\}$.
    while $t < 1$ **do**
      Identify top item $top^i(j)$ in type i for each agent $j \in N^i$.
      Consume.
        For each $o \in D_i$,$consumers(o) \leftarrow |\{j \in N^i | top^i(j) = o\}|$.
        $\rho \leftarrow \min_{o\in D_i} \frac{supply(o)}{consumers(o)}$.
        For each $j \in N^i$, $p^i_{j,top^i(j)} \leftarrow p^i_{j,top^i(j)} + \rho$.
        For each $o \in D_i$, $supply(o) \leftarrow supply(o) - \rho \cdot consumers(o)$.
        $t \leftarrow t + \rho$.
For each $j \leq n, \mathbf{x} \in \mathscr{D}, p_{j,\mathbf{x}} = \prod_{o=D_i(\mathbf{x}), i\leq d} p^i_{j,o}$.
return P

In the LexiPS mechanism, agent j always consumes her most preferred item o_j with positive supply in the current most important type. Agent j consumes o_j until one of the following occurs:

(i) There is no supply of o_j left, after which agent j stops consuming o_j and starts to consume her most preferred item according to $\succ^i_j$ that is currently with positive supply.
(ii) $\sum_{o\in D_i} p^i_{j,o} = 1, o_j \in D_i$, after which agent j turns to her next most important type according to $\rhd_j$ and starts consumes her favorite item that is with positive supply of that type.

After consumption, we obtain P^i for each type $i \leq d$. By the construction, the allocation for each type is made independently, and therefore we construct the assignment matrix P by computing each element as follows:

$$p_{j,\mathbf{x}} = \prod_{o=D_i(\mathbf{x}), i\leq d} p^i_{j,o}. \tag{2}$$

LexiPS runs in d phases. In each phase, each agent j identifies current most important type i_j and only consumes items of type i_j. The time t for each phase is one unit. At the beginning of each phase, we set the timer $t = 0$. During the consumption, agent $j \in N_i$, where N_i is set of agents whose current most importance type is i, first decides her most preferred *unexhausted* item $top^i(j)$ of type i according to $\succ^i_j$. Here we say that an item o is exhausted if the supply $supply(o) = 0$. Agent j consumes the item $top^i(j)$ at a uniform rate of one unit per unit of time. The consumption pauses whenever one of the items being consumed becomes exhausted. That means agent j's share of $top^i(j)$ is increased by ρ, the duration since last pause, and the supply of item o, i.e. $supply(o)$ is computed by subtracting ρ for $consumers(o)$ times, the number of agents j such that $top^i(j) = o$. In Algorithm 1, ρ is computed as $\min_{o\in M} \frac{supply(o)}{consumers(o)}$. After this, we increase the timer t by ρ, identify $top^i(j)$ for each agent, and continue the consumption. The current phase ends when the timer t reaches 1, and the algorithm starts the next phase.

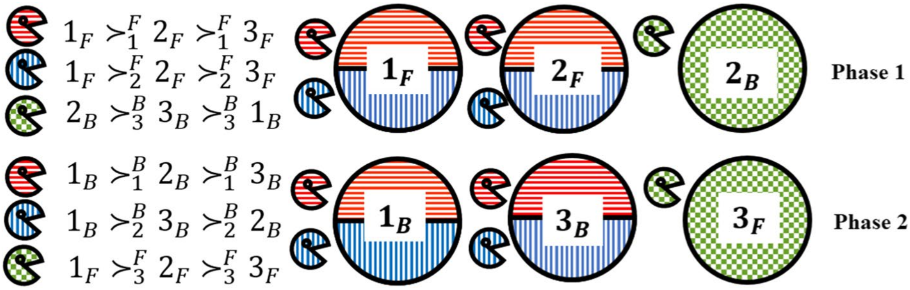


Fig. 1 Execution of LexiPS in Example 4

We demonstrate how LexiPS outputs decomposable assignments that also satisfy sd-efficiency and sd-envy-freeness using a simple example first, by describing the execution of LexiPS on the MTRA from Example 2 where agents have lexicographic preferences. In the first phase, agent 1 picks her most preferred item 1_F of her most important type F, and agent 2 picks 1_B of her most important type B. In the second phase, agents 1 and 2 can only pick the remaining items to meet their demand of one bundle consisting of one unit of each type, i.e. 2_B and 2_F, respectively. Therefore, LexiPS outputs the following assignment:

Agent	1_F1_B	1_F2_B	2_F1_B	2_F2_B
1	0	1	0	0
2	0	0	1	0

It is easy to check that the output of LexiPS for this MTRA is decomposable, sd-efficient, and sd-envy-free. In Theorem 2, we show that LexiPS always outputs decomposable assignments that satisfy sd-efficiency and sd-envy-freeness $R \in \mathscr{R}$.

We illustrate the execution of LexiPS further in Example 4.

Example 4 Consider an MTRA (N, M) where $N = \{1, 2, 3\}$, $M = D_F \times D_B, D_F = \{1_F, 2_F, 3_F\}, D_B = \{1_B, 2_B, 3_B\}$, and the profile $R = \{\succ_1, \succ_2, \succ_3\}$. The preferences $\succ_1, \succ_2, \succ_3$ are as follows:

Agent	Preferences
1	$F \rhd_1 B, 1_F \succ_1^F 2_F \succ_1^F 3_F, 1_B \succ_1^B 2_B \succ_1^B 3_B$
2	$F \rhd_2 B, 1_F \succ_2^F 2_F \succ_2^F 3_F, 1_B \succ_2^B 3_B \succ_2^B 2_B$
3	$B \rhd_3 F, 1_F \succ_3^F 2_F \succ_3^F 3_F, 2_B \succ_3^B 3_B \succ_3^B 1_B$

The execution of LexiPS is shown in Fig. 1. In Phase 1, agents 1 and 2 consume items in D_F, while agent 3 consumes alone in D_B. Therefore, agent 3 gets her most preferred items 2_B in D_B fully, and 1_B and 3_B are left. Since agents 1 and 2 have the same preference for D_F, each of them obtains 0.5 units of 1_F and 0.5 units of 2_F, and 3_F is left. Similarly in Phase 2, agents 1 and 2 prefer type B while agent 3 prefers F. Then agent 3 gets the remaining item 3_F, and agents 1 and 2 divide 1_B and 3_B uniformly according to their preferences. The following table shows agents' allocations of items after each phase:

Agent	Phase 1	Phase 2
1	0.5 of 1_F, 0.5 of 2_F	0.5 of 1_B, 0.5 of 3_B
2	0.5 of 1_F, 0.5 of 2_F	0.5 of 1_B, 0.5 of 3_B
3	1 of 2_B	1 of 3_F

According to line 11 of Algorithm 1, the output is the multi-type assignment P below (To save space, from here on, we omit columns corresponding to bundles for which every agent receives 0 share.):

Agent	P				
	1_F1_B	1_F3_B	2_F1_B	2_F3_B	3_F2_B
1	0.25	0.25	0.25	0.25	0
2	0.25	0.25	0.25	0.25	0
3	0	0	0	0	1

It is easy to check that P is decomposable. □

4.2 Properties of LexiPS

In this subsection, we show in Theorem 2 that similarly to PS, LexiPS satisfies properties of efficiency and envyfreeness based on our extension of stochastic dominance for MTRAs under lexicographic preferences, and additionally, the output of LexiPS is always decomposable and therefore can be applied to MTRAs with indivisible items.

Before we begin, we introduce some notations for convenience. Given a fractional allocation p, we define p^i to be the fractional allocation of items of type i as an n-vector with each component corresponding to an item $o \in D_i$, whose value is $p^i_o = \sum_{o\in\mathbf{x},\mathbf{x}\in\mathcal{D}} p_{\mathbf{x}}$ representing the total shares of bundles containing o. We also define a *partial bundle* containing a single item of each type in any set of types $H \subseteq \{i|i \le d\}$ to be the vector $\mathbf{w} \in \prod_{i\in H} D_i$. For any (partial) bundle $\mathbf{w}$, we use $w_i = D_i(\mathbf{w})$ to denote the item of type i in the bundle $\mathbf{w}$. For any partial bundle $\mathbf{w} \in \prod_{i\in H} D_i$, we define $Z_{\mathbf{w}}$ as the set of bundles which contain all the items in $\mathbf{w}$, i.e. $Z_{\mathbf{w}} = \{\mathbf{x} \in \mathcal{D} | \text{for all } i \in H, x_i = w_i\}$. We use o_i to refer to an item o of type i to make the type of the item clear in the exposition. To show the items in a (partial) bundle with items from the types in H directly, we use $(o_i)_{i\in H}$ to denote the bundle containing items o_i of each type $i \in H$. W.l.o.g. let $Z_{()} = \mathcal{D}$, where $()$ is the partial bundle which does not contain any items.

Before proving Theorem 2, we provide Lemma 1 which is useful for comparing two allocations over an upper contour set. The full proof of Lemma 1 is in "Proof of Lemma 1" in Appendix.

Lemma 1 *Let $\succ$ be any lexicographic preference relation with importance order $1 \rhd 2 \cdots \rhd d$, and p be a fractional allocation where $p_{\mathbf{x}} = \prod_{h\le d} p^h_{x_h}$. Let q be a factional allocation which satisfies one of the following conditions:*

- (i) *$q_{\mathbf{x}} = \prod_{h\le d} q^h_{x_h}$, and there exists a type $i \le d$ such that $p^h_o = q^h_o$ for any $h < i$.*
- (ii) *there exists a type $i \le d$ such that $\sum_{\mathbf{x}\in Z_{\mathbf{w}}} p_{\mathbf{x}} = \sum_{\mathbf{x}\in Z_{\mathbf{w}}} q_{\mathbf{x}}$ for any $h < i$ and $\mathbf{w} \in D_1 \times \cdots \times D_h$.*

Then, for any bundle $\mathbf{y}$, *it holds that*

$$\sum_{\mathbf{x}\in U(\succ,\mathbf{y})\cap Z_{(y_h)_{h\leq i-1}}} p_{\mathbf{x}} \geq \sum_{\mathbf{x}\in U(\succ,\mathbf{y})\cap Z_{(y_h)_{h\leq i-1}}} q_{\mathbf{x}}, \tag{3}$$

if and only if $\sum_{\mathbf{x}\in U(\succ,\mathbf{y})} p_{\mathbf{x}} \geq \sum_{\mathbf{x}\in U(\succ,\mathbf{y})} q_{\mathbf{x}}$.

Example 5 We illustrate Lemma 1 with the allocation q which satisfies the condition (ii) of Lemma 1. Given an MTRA with $D_F = \{1_F, 2_F\}$ and $D_B = \{1_B, 2_B\}$, let $\succ$ satisfy the conditions $F \rhd B$, $1_F \succ^F 2_F$, and $1_B \succ^B 2_B$. Suppose that for type F, the allocations p and q satisfy that $\sum_{\mathbf{x}\in Z_{(1_F)}} p_{\mathbf{x}} = \sum_{\mathbf{x}\in Z_{(1_F)}} q_{\mathbf{x}}$ and $\sum_{\mathbf{x}\in Z_{(2_F)}} p_{\mathbf{x}} = \sum_{\mathbf{x}\in Z_{(2_F)}} q_{\mathbf{x}}$. Let $\mathbf{y} = 2_F 1_B$ and we have that $\{\mathbf{x} | D_F(\mathbf{x}) \succ^F 2_F\} = Z_{(1_F)}$ and $Z_{(2_F,1_B)} \cap U(\succ, \mathbf{y}) = \{\mathbf{y}\}$. Then, if $\sum_{\mathbf{x}\in U(\succ,\mathbf{y})} p_{\mathbf{x}} \geq \sum_{\mathbf{x}\in U(\succ,\mathbf{y})} q_{\mathbf{x}}$, it holds that $p_{\mathbf{y}} \geq q_{\mathbf{y}}$ because

$$\sum_{\mathbf{x}\in U(\succ,\mathbf{y})} p_{\mathbf{x}} = \sum_{\mathbf{x}\in Z_{(1_F)}} p_{\mathbf{x}} + p_{\mathbf{y}} = \sum_{\mathbf{x}\in Z_{(1_F)}} q_{\mathbf{x}} + p_{\mathbf{y}} \geq \sum_{\mathbf{x}\in Z_{(1_F)}} q_{\mathbf{x}} + q_{\mathbf{y}} = \sum_{\mathbf{x}\in U(\succ,\mathbf{y})} q_{\mathbf{x}}.$$

With this, we can also prove that $\sum_{\mathbf{x}\in U(\succ,\mathbf{y})} p_{\mathbf{x}} \geq \sum_{\mathbf{x}\in U(\succ,\mathbf{y})} q_{\mathbf{x}}$ if $p_{\mathbf{y}} \geq q_{\mathbf{y}}$. □

With Lemma 1, we show the three properties of LexiPS in the following Theorem 2.

Theorem 2 *For MTRAs with lexicographic preferences, LexiPS satisfies sd-efficiency and sd-envy-freeness. Especially, LexiPS outputs decomposable assignments.*

Proof Given an MTRA (N, M) and profile R of lexicographic preferences, let $P = LexiPS(R)$. For ease of exposition, we divide the proof into three parts, one each to show that LexiPS satisfies sd-efficiency, sd-envy-freeness, and is guaranteed to output decomposable assignments respectively.

Part 1 [*sd-efficiency*] Suppose for the sake of contradiction, we suppose that there exists an assignment $Q \neq P$ such that $Q \succeq^{sd} P$. Then, for any agent $k \in N$, $Q \succeq^{sd}_k P$, and there exists an agent j who strictly prefers her allocation in Q to the one in P, i.e. $Q_j \succeq^{sd}_j P_j$ and $Q_j \neq P_j$. W.l.o.g, let the types be labeled according to $\rhd_j$ as $1 \rhd_j 2 \rhd_j \cdots \rhd_j d$.

We show that $Q_j = P_j$ by proving the following equation by mathematical induction on the types: for any $i \leq d$ and $o_1, \ldots, o_i$,

$$\sum_{\mathbf{x}\in Z_{(o_h)_{h\leq i}}} p_{j,\mathbf{x}} = \sum_{\mathbf{x}\in Z_{(o_h)_{h\leq i}}} q_{j,\mathbf{x}}. \tag{4}$$

Base case We prove the Eq. (4) for $i = 1$, which is equivalent to $Q^1_j = P^1_j$. First we show that $Q^1 \succeq^{sd}_j P^1$. Suppose it is false, and then there must exist an item y_1 and the least preferred bundle $\mathbf{y}$ containing y_1 w.r.t. $\succ_j$ such that

$$\sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}} = \sum_{o\in U(\succ^1_j,y_1)} p_{j,o} > \sum_{o\in U(\succ^1_j,y_1)} q_{j,o} = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} q_{j,\mathbf{x}}.$$

This is a contradiction to our assumption that $Q \succeq^{sd}_j P$. Having shown that $Q^1 \succeq^{sd}_j P^1$, our claim that $Q^1_j = P^1_j$ follows from Claim 1 below. We provide the proof of Claim 1 in "Proof of Claim 1 in Theorem 2" in Appendix.

Claim 1 Given an MTRA (N, M) and a lexicographic preference profile R, let $P = LexiPS(R)$ and Q be an assignment such that there exists $i \le d$ such that $Q_j^{h_j} = P_j^{h_j}$ for any $h < i$ and agent j with the importance order $1_j \rhd_j 2_j \rhd_j \cdots \rhd_j d_j$. Then, $Q_j^{i_j} = P_j^{i_j}$ if $Q^{i_j} \succeq_j^{sd} P^{i_j}$ for any agent j.

Inductive step Now, we prove the Eq. (4) for type $1 < i \le d$ using Lemma 1 and Claim 1. Assume that for any $h < i$ and items $o_1, \dots, o_h$, the total shares of bundles containing these items are equal in P and Q:

$$\sum_{\mathbf{x} \in Z_{(o_g)_{g \le h}}} p_{j,\mathbf{x}} = \sum_{\mathbf{x} \in Z_{(o_g)_{g \le h}}} q_{j,\mathbf{x}}. \tag{5}$$

First we show that $Q^i \succeq_j^{sd} P^i$. For any $h \le i$, let y_h be an arbitrary item of type h. W.l.o.g, let $\mathbf{y}$ be the least preferred bundle in $Z_{(y_h)_{h \le i}}$ w.r.t. $\succ_j$. Let $S = \{\mathbf{x} \in Z_{(y_h)_{h \le i-1}} | x_i \in U(\succ_j^i, y_i)\}$. Then, for any $\mathbf{x} \in S$, we have that $\mathbf{x} \in U(\succ_j, \mathbf{y})$, which also means that

$$\sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} = \sum_{\mathbf{x} \in Z_{(y_h)_{h \le i-1}} \cap U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}}, \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}} = \sum_{\mathbf{x} \in Z_{(y_h)_{h \le i-1}} \cap U(\succ_j, \mathbf{y})} q_{j,\mathbf{x}}. \tag{6}$$

By the assumption that $Q \succeq_j^{sd} P$, we have that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} \le \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} q_{j,\mathbf{x}}$. With this and the Eq. (5), we have that $\sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} \le \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}}$ by Lemma 1. Recall that y_h is an arbitrarily chosen item in D_h. By summing up each side over all the possible choices of $y_1, \dots, y_{i-1}$, we have that

$$\sum_{y_1 \in D_1} \cdots \sum_{y_{i-1} \in D_{i-1}} \sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} \le \sum_{y_1 \in D_1} \cdots \sum_{y_{i-1} \in D_{i-1}} \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}}. \tag{7}$$

After simplifying, the inequality (7) means that for any $y_i \in D_i$,

$$\sum_{o \in U(\succ_j^i, y_i)} p_{j,o}^i = \sum_{\mathbf{x} \in \{\mathbf{x} | x_i \in U(\succ_j^i, y_i)\}} p_{j,\mathbf{x}} \le \sum_{\mathbf{x} \in \{\mathbf{x} | x_i \in U(\succ_j^i, y_i)\}} q_{j,\mathbf{x}} = \sum_{o \in U(\succ_j^i, y_i)} q_{j,o}^i,$$

which implies that $Q^i \succeq_j^{sd} P^i$.

Then by Claim 1, we have that $Q_j^i = P_j^i$. We have already shown that $\sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} \le \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}}$ for $\mathbf{y}$ with an arbitrary choice of y_h for each $h \le i$, and we now claim that $\sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} = \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}}$. Otherwise, if $\sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} < \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}}$ for some $\mathbf{y}$ with a certain choice of y_h for each $h \le i$, then by the inequality (7) we must have that $\sum_{o \in U(\succ_j^i, y_i)} p_{j,o}^i < \sum_{o \in U(\succ_j^i, y_i)} q_{j,o}^i$, a contradiction to $Q_j^i = P_j^i$. Therefore, we have that $\sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} = \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}}$, which is equivalent to the Eq. (4) for type i.

By mathematical induction, we show that the Eq. (4) holds true for any $i \le d$, which means that $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$, a contradiction to our assumption that $Q_j \ne P_j$. This completes the proof.

Part 2 [*sd-envy-freeness*] The following proof involves tracking the execution of LexiPS phase by phase one after the other. In each phase, every agent spends one unit of time consuming items of one type in LexiPS. We first prove that no agent j envies another agent who has the same importance order. For convenience, we label the types according to $\rhd_j$. Let N_i be the set of agents who consume items of types i in Phase i. The execution of LexiPS in Phase i can be viewed as PS for the single type allocation problem with agents in N_i and available items left in D_i in Phase i. By [9], we know that PS satisfies sd-envy-freeness. Therefore, we have that for any agent $k \in N_i$, $\sum_{o' \in U(\succ_j^i, o)} p_{j,o'}^i \ge \sum_{o' \in U(\succ_j^i, o)} p_{k,o'}^i$ for any

$o \in D_i$, i.e. $P^i_j \succeq^{sd}_j P^i_k$. With this, it follows from Claim 2 that $P_j \succeq^{sd}_j P_k$. We provide the proof of Claim 2 in "Proof of Claim 2 in Theorem 2" in Appendix.

Claim 2 Given a lexicographic preference relation $\succ$ and two factional allocations p and q which satisfy $p_{\mathbf{x}} = \prod_{i \le d, o = D_i(\mathbf{x})} p^i_o$ and $q_{\mathbf{x}} = \prod_{i \le d, o = D_i(\mathbf{x})} q^i_o$ respectively, if $p^i \succeq^{sd} q^i$ for type i and $p^h = q^h$ for any $h \neq i$, then we have that $p \succeq^{sd} q$.

Now, we prove that agent j does not envy agents who have different importance orders. Assume for the sake of contradiction that there exists such an agent k and $P_j \not\succeq^{sd}_j P_k$, i.e. there exists $\mathbf{y} \in \mathscr{D}$ which satisfies

$$\sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} < \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{k,\mathbf{x}}. \tag{8}$$

Because agent k has a different importance order from agent j, by construction of LexiPS, there must be a Phase in *LexiPS* where agents j and k consume items of different types. We show that this contradicts the assumption (8). W.l.o.g. let Phase i be the earliest phase in the execution of *LexiPS*(R) where j and k consume items of different types. It follows that $k \notin N_i$, and by the selection of i, it must hold that $k \in N_h$ for any $h < i$. Then, by sd-envy-freeness of PS, $P^h_j \succeq^{sd}_j P^h_k$ for any $h < i$. With this and Claim 2, given an allocation q with $q_{\mathbf{x}} = \prod_{i \le d} q^i_{x_i}$ where $q^h = P^h_j \succeq^{sd}_j P^h_k$ for $h < i$ and $q^g = P^g_k$ for any $g \ge i$, we have that $q \succeq^{sd}_j P_k$. Therefore, we see that if $P_j \succeq^{sd}_j q$, then $P_j \succeq^{sd}_j P_k$. We show that $P_j \succeq^{sd}_j q$ in the following.

By the selection of i, it must hold that agent k consumes items of type i in a phase that comes strictly after phase i where agent j consumes items of type i. Then, for any pair of items y_i and z_i such that $p^i_{j,y_i} > 0$ and $q^i_{z_i} = p^i_{k,z_i} > 0$ respectively, it must hold that either $y_i \succ_j z_i$ or $y_i = z_i$ because the unexhausted items of type i at the end of Phase i are not preferred to those consumed by agent j in Phase i w.r.t. $\succ^i_j$.

Case (i) Suppose that $y_i \succ_j z_i$ for any y_i and z_i with $p^i_{j,y_i} > 0$ and $q^i_{z_i} > 0$, respectively. W.l.o.g. let y_i be the least preferred item w.r.t. $\succ^i_j$ with $p^i_{j,y_i} > 0$, and therefore it follows that $\sum_{o \in U(\succ^i_j, y_i)} p^i_{j,o}$=1. Due to the fact that $P^h_j = q^h$ for $h < i$, we know that for any bundle $\mathbf{w}$,

$$\sum_{\mathbf{x} \in Z_{(w_h)_{h \le i-1}}} P_{j,\mathbf{x}} = \prod_{h<i} p^h_{j,w_h} = \prod_{h<i} q^h_{w_h} = \sum_{\mathbf{x} \in Z_{(w_h)_{h \le i-1}}} q_{\mathbf{x}}.$$

Then, for $\mathbf{w}$ with $w_i \in U(\succ^i_j, y_i)$, we have that $q_{\mathbf{w}} = 0$. For $\mathbf{w}$ with $y_i \succ^i_j w_i$,

$$\sum_{\mathbf{x} \in Z_{(w_h)_{h \le i-1}} \cap U(\succ_j, \mathbf{w})} P_{j,\mathbf{x}} = \prod_{h<i} p^h_{j,w_h} \cdot \sum_{o \in U(\succ^i_j, y_i)} p^i_{j,o} = \sum_{\mathbf{x} \in Z_{(w_h)_{h \le i-1}}} P_{j,\mathbf{x}} \ge \sum_{\mathbf{x} \in Z_{(w_h)_{h \le i-1}} \cap U(\succ_j, \mathbf{w})} q_{\mathbf{x}}.$$

Together, they imply that for any $\mathbf{w} \in \mathscr{D}$,

$$\sum_{\mathbf{x} \in Z_{(w_h)_{h \le i-1}} \cap U(\succ_j, \mathbf{w})} p_{j,\mathbf{x}} \ge \sum_{\mathbf{x} \in Z_{(w_h)_{h \le i-1}} \cap U(\succ_j, \mathbf{w})} q_{\mathbf{x}} \tag{9}$$

It follows from the inequality (9), the fact that $P^h_j = q^h$ for every $h < i$, and Lemma 1 that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{w})} p_{j,\mathbf{x}} \ge \sum_{\mathbf{x} \in U(\succ_j, \mathbf{w})} q_{\mathbf{x}}$ for any $\mathbf{w}$, a contradiction to the assumption in Eq. (8).

Case (ii) Suppose that there exist y_i such that $p^i_{j,y_i} > 0$ and $q^i_{y_i} > 0$. It is easy to see from the construction of LexiPS that y_i is the least preferred item consumed by agent j

according to $\succ_j^i$, and also the most preferred item consumed by agent k according to $\succ_j^i$. Then we have that

$$\sum_{o \succ_j^i y_i} p_{j,o}^i = 1 - p_{j,y_i}^i \geq q_{y_i}^i = \sum_{o \in U(\succ_j^i, y_i)} q_o^i. \tag{10}$$

Let $\mathbf{y}$ be an arbitrary bundle containing y_i. Then, from the Eq. (2) which computes the shares of bundles assigned by LexiPS, we define $\alpha = \prod_{h<i} p_{j,y_h}^i = \prod_{h<i} q_{y_h}^i$ and it follows that

$$\sum_{\mathbf{x} \in Z_{(y_h)_{h \leq i-1}} \cap U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} \geq \alpha \cdot \sum_{o \succ_j^i y_i} p_{j,o}^i \geq \alpha \cdot \sum_{o \in U(\succ_j^i, y_i)} q_o^i \geq \sum_{\mathbf{x} \in Z_{(y_h)_{h \leq i-1}} \cap U(\succ_j, \mathbf{y})} q_{\mathbf{x}}. \tag{11}$$

By the inequality (11), the fact that $P_j^h = q^h$ for every $h < i$, and Lemma 1, we have that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} \geq \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} q_{\mathbf{x}}$. For any other bundle $\mathbf{w}$ such that $w_i \neq y_i$, by using a similar argument to Case (i), we have that $q_{\mathbf{w}} = 0$ if $w_i \in U(\succ_j^i, y_i)$ and $\sum_{\mathbf{x} \in Z_{(w_h)_{h \leq i-1}} \cap U(\succ_j, \mathbf{w})} P_{j,\mathbf{x}} \geq \sum_{\mathbf{x} \in Z_{(w_h)_{h \leq i-1}} \cap U(\succ_j, \mathbf{w})} q_{\mathbf{x}}$ if $y_i \succ_j^i w_i$, and it holds that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{w})} p_{j,\mathbf{x}} \geq \sum_{\mathbf{x} \in U(\succ_j, \mathbf{w})} q_{\mathbf{x}}$.

Together, we have that $P_j \succeq_j^{sd} q$ and therefore $P_j \succeq_j^{sd} P_k$, which is a contradiction to the assumption in Eq. (8). This means that agent j does not envy any other agent who has a different importance order. Together with our earlier conclusion that agent j does not envy any other agents who has the same importance order, we conclude that agent j does not envy any other agent.

Part 3 [*decomposable output*] Let $A = (a_{j,\mathbf{x}})_{j \in N, \mathbf{x} \in \mathcal{D}}$ be a multi-type discrete assignment in $\mathcal{A}$, and we use $A^i = (a_{j,o}^i)_{j \in N, o \in D_i}$ to refer to the single type discrete assignment for each $i \leq d$ where $a_{j,o_i}^i = \sum_{o_i \in \mathbf{x}} a_{j,\mathbf{x}}$ for each $o_i \in D_i$. Before we begin the proof, we show the fact that any collection of d discrete assignments $(A^i)_{i \leq d}$ determines a unique multi-type discrete assignment A, because for any $j \leq n$ and $\mathbf{x} \in \mathcal{D}$, agent j is assigned the bundle $\mathbf{x}$ if she is assigned all the items in $\mathbf{x}$, i.e.

$$a_{j,\mathbf{x}} = \prod_{i \leq d} a_{j,x_i}^i. \tag{12}$$

We provide Example 6 to show this relation between $(A^i)_{i \leq d}$ and A.

Recall that $P = LexiPS(R)$ and P^i refers to the single type discrete assignment for each $i \leq d$. Let $\mathcal{A}^i$ be the set of all the discrete assignments of the n items of type i to n agents. Then by the Birkhoff-Von Neumann theorem, P^i describes a probability distribution over $\mathcal{A}^i$. Consider an arbitrary fixed distribution over $\mathcal{A}^i$ described by P^i where α^{A^i} is the probability associated with A^i. It follows that $P^i = \sum_{A^i \in \mathcal{A}^i} \alpha^{A^i} \cdot A^i$ for any $i \leq d$. Then from the Eq. (2) which computes the shares of bundles assigned by LexiPS, it holds that for any $\mathbf{x} \in \mathcal{D}$,

$$p_{j,\mathbf{x}} = \prod_{i \leq d} p_{j,x_i}^i = \prod_{i \leq d} \sum_{A^i \in \mathcal{A}^i} \alpha^{A^i} \cdot a_{j,x_i}^i \tag{13}$$

The result of the Eq. (13) is a product of d polynomials, and we can rewrite it as one polynomial as follows:

$$\begin{aligned} p_{j,\mathbf{x}} &= \sum_{A^1\in\mathscr{A}^1, A^2\in\mathscr{A}^2,\dots,A^d\in\mathscr{A}^d} \prod_{i\le d}(\alpha^{A^i}\cdot a^i_{j,x_i}) \\ &= \sum_{A^1\in\mathscr{A}^1, A^2\in\mathscr{A}^2,\dots,A^d\in\mathscr{A}^d} \left(\prod_{i\le d}\alpha^{A^i}\cdot\prod_{i\le d} a^i_{j,x_i}\right). \end{aligned}$$

Let A be the unique multi-type discrete assignment determined by any collection of $(A^i)_{i\le d}$ and $\alpha^A = \prod_{i\le d}\alpha^{A^i}$. Recall that α^{A^i} is the probability of A^i which follows the distribution described by P^i. With the Eq. (12) we further have that

$$p_{j,\mathbf{x}} = \sum_{A\in\mathscr{A}} \alpha^A \cdot a_{j,\mathbf{x}}.$$

It is easy to see that α^A can be viewed as the probability of A following some distribution over $\mathscr{A}$ which can be described by P. It means that $P = \sum_{A\in\mathscr{A}}\alpha^A\cdot A$ and therefore P is decomposable. □

Example 6 Given an MTRA with types F and B and agents 1 and 2, we show the collection of the following single type discrete assignments A^F and A^B corresponds to a unique multi-type discrete assignment A:

Agent	A^F		A^B		Agent	A			
	1_F	2_F	1_B	2_B		1_F1_B	1_F2_B	2_F1_B	2_F2_B
1	1	0	0	1	1	0	1	0	0
2	0	1	1	0	2	0	0	1	0

For example, we see that agent 1 obtains item 1_F in A^F and item 2_B in A^B, and therefore she obtains the bundle 1_F2_B in assignment A accordingly, i.e. $a_{1,1_F2_B} = a^F_{1,1_F}\cdot a^B_{1,2_B} = 1$. □

Remark 2 LexiPS does not satisfy lexi-efficiency. To show this, we revisit Example 4 and compare the assignment P in Example 4 with the assignment Q below.

Agent	Q				
	1_F1_B	1_F3_B	2_F1_B	2_F3_B	3_F2_B
1	0.5	0	0	0.5	0
2	0.25	0.25	0.25	0.25	0
3	0	0	0	0	1

For agent 1, her allocation in P is lexicographically dominated by the one in Q. We also note that in Q, agents 2 and 3 obtain the same allocations as in P. Therefore, we have that $Q \succ^{lexi} P$. □

As we show in Remark 3, LexiPS is not sd-weak-strategyproof, similarly to other extensions of PS [3, 25, 28, 45] which also do not satisfy sd-weak-strategyproofness. Theorem 3 shows an exception that LexiPS is able to satisfy sd-weak-strategyproofness if we add the restriction that agents cannot misreport their importance orders over the types.

Theorem 3 *For MTRAs with lexicographic preferences, LexiPS satisfies sd-weak-strategyproofness when agents report importance orders truthfully.*

Proof Suppose that agent j misreports her preference for some types and obtains a better allocation. Let $\succ'_j$ be her new preference after misreporting and $R' = (\succ'_j, \succ_{-j})$ be the resulting preference profile. Let $P = LexiPS(R)$ and $Q = LexiPS(R')$. From the execution of LexiPS, we observe that each single type assignment generated only depends on the preferences of all of the agents over items of the corresponding type. Therefore, for any type i where agent j does not misreport her preference, her allocation remains unchanged in Q^i, i.e. $Q^i_j = P^i_j$. It means that if $Q^i_j \neq P^i_j$, then agent j must misreport her preference of type i.

For convenience, let the types be labeled according to $\rhd_j$. Let i be the most important type where j has a different allocation in Q, i.e. $Q^i_j \neq P^i_j$. Then by our assumption of the misreport being beneficial for agent j, we have that $Q \succeq^{sd}_j P$, $Q \neq P$ and that for any $h < i$, $P^h_j = Q^h_j$. The phase when agent j consumes items in D_i can be viewed as executing PS on type i. From [9], we know that PS satisfies sd-weak-strategyproofness, which means $Q^i_j = P^i_j$ if $Q^i \succeq^{sd}_j P^i$. Because $Q^i_j \neq P^i_j$, we have that $Q^i \not\succeq^{sd}_j P^i$, which also means that there exists y_i such that

$$\sum_{o \in U(\succ^i_j, y_i)} p^i_{j,o} > \sum_{o \in U(\succ^i_j, y_i)} q^i_{j,o}. \tag{14}$$

Let $\mathbf{y}$ be the bundle with such an item y_i and for $h < i$, y_h is an item of type h satisfying $p^h_{j,y_h} \neq 0$, which also means $q^h_{j,y_h} = p^h_{j,y_h} \neq 0$. W.l.o.g. let $\mathbf{y}$ be the least preferred bundle in $Z_{(y_h)_{h \leq i}}$. Let $S = \{\mathbf{x} \in Z_{(y_h)_{h\leq i-1}} | x_i \in U(\succ^i_j, y_i)\}$. By the inequality (14) and our observation that $P^h_j = Q^h_j$ for any $h < i$, we have that

$$\begin{aligned}\sum_{\mathbf{x} \in \{\mathbf{x} \in Z_{(y_h)_{h\leq i-1}} | x_i \in U(\succ^i_j, y_i)\}} p_{j,\mathbf{x}} = \sum_{\mathbf{x} \in S} p_{j,\mathbf{x}} = \prod_{h<i} p^h_{j,y_h} \cdot \sum_{o \in U(\succ^i_j, y_i)} p^i_{j,o} > \\ \prod_{h<i} q^h_{j,y_h} \cdot \sum_{o \in U(\succ^i_j, y_i)} q^i_{j,o} = \sum_{\mathbf{x} \in S} q_{j,\mathbf{x}} = \sum_{\mathbf{x} \in \{\mathbf{x} \in Z_{(y_h)_{h\leq i-1}} | x_i \in U(\succ^i_j, y_i)\}} q_{j,\mathbf{x}}.\end{aligned} \tag{15}$$

With the assumption that $P^h_j = Q^h_j$ for any $h < i$ and the inequality (15), by Lemma 1, we have that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} > \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} q_{j,\mathbf{x}}$, which means agent j does not obtain a better allocation in Q, a contradiction. Therefore, we have that LexiPS is sd-weak-strategyproof when agents report importance order truthfully. □

Remark 3 When applying LexiPS to MTRAs with lexicographic preferences, an agent may get a better allocation by misreporting her importance order. Consider an MTRA with lexicographic preferences where there are agents 1 and 2 and types F, B and T. Both agents prefer 1_i to 2_i for $i \in \{F, B, T\}$, but their preferences over bundle are different due to their importance orders as follows:

Agent	Importance Order
1	$F \rhd_1 B \rhd_1 T$
2	$T \rhd_2 F \rhd_2 B$

LexiPS gives the fractional assignment denoted by P. If agent 2 misreports her importance order as $\rhd'_2$: $F \rhd'_2 T \rhd'_2 B$, LexiPS gives another fractional assignment denoted by P'. Both P and P' are shown as follows:

Agent	P		Agent	P'			
	$1_F 1_B 2_T$	$2_F 2_B 1_T$		$1_F 1_B 2_T$	$1_F 2_B 1_T$	$2_F 1_B 2_T$	$2_F 2_B 1_T$
1	1	0	1	0.5	0	0.5	0
2	0	1	2	0	0.5	0	0.5

We observe that compared with P, agent 2 loses 0.5 shares of $2_F 2_B 1_T$, but acquires 0.5 shares of $1_F 2_B 1_T$ in P'. Since $1_F 2_B 1_T \succ_2 2_F 2_B 1_T$, we obtain that $P' \succeq_2^{sd} P$, but $P \succeq_2^{sd} P'$ is false, which means that LexiPS does not satisfy sd-weak-strategyproofness when an agent can misreport her importance order. □

5 MPS for MTRAs with divisible items

In this section we consider MTRAs with divisible items under the unrestricted domain of strict linear preferences. We present a simplified version of the MPS mechanism proposed by [42] in Algorithm 2, since we do not need to deal with partial preferences. At a high level, in MPS agents consume bundles consisting of d items, one of each type, in contrast with PS where agents consume items directly. We prove in Theorem 4 that under strict linear preferences, MPS satisfies lexi-efficiency, which is a stronger notion of efficiency, and implies sd-efficiency, and prove in Proposition 3 that MPS also satisfies sd-envy-freeness under strict linear preferences. In Theorem 6, we show that MPS also satisfies sd-weak-strategyproofness under the domain restriction of lexicographic preferences. In addition, we also provide two separate characterizations of MPS involving leximin-optimality and item-wise ordinal fairness in Theorem 7.

5.1 The MPS mechanism

Given an MTRA (N, M) and a preference profile $R = (\succ_j)_{j \leq n}$, MPS proceeds in multiple rounds as follows: At the beginning of each round, M' contains all the items that are unexhausted. Each agent j first decides her most preferred *available* bundle $top(j)$ according to $\succ_j$. A bundle $\mathbf{x}$ is available so long as every item $o \in \mathbf{x}$ is unexhausted. Then, each agent consumes their most preferred available bundle by consuming all of the items in it at a uniform rate of one unit per unit of time. The round ends whenever one of the bundles being consumed becomes unavailable because an item being consumed has been exhausted. The algorithm terminates when all the items are exhausted.

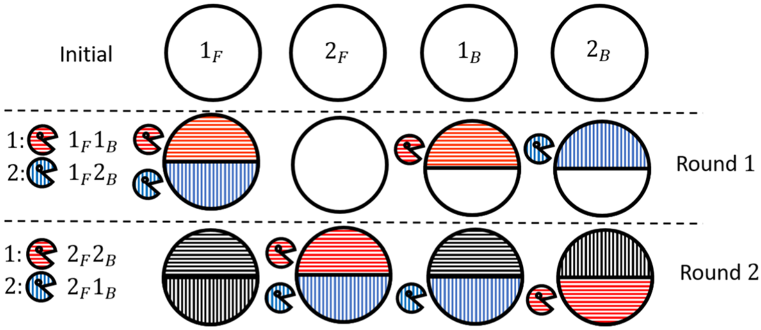


Fig. 2 An example of the execution of MPS

Algorithm 2 MPS for MTRAs under strict linear preferences.

1: **Input:** An MTRA (N, M) and a preference profile R.
2: For each $o \in M$, $supply(o) \leftarrow 1$. $M' \leftarrow M$. $P \leftarrow 0^{n \times |\mathscr{D}|}$.
3: **while** $M' \neq \emptyset$ **do**
4: **Identify top bundle** $top(j)$ for each agent $j \leq n$.
5: **Consume.**
5.1: For any $o \in M'$, $consumers(o) \leftarrow |\{j \in N | o \in top(j)\}|$.
5.2: $\rho \leftarrow \min_{o \in M'} \frac{supply(o)}{consumers(o)}$.
5.3: For each $j \leq n$, $p_{j,top(j)} \leftarrow p_{j,top(j)} + \rho$.
5.4: For each $o \in M'$, $supply(o) \leftarrow supply(o) - \rho \cdot consumers(o)$.
6: $B \leftarrow \arg\min_{o \in M'} \frac{supply(o)}{consumers(o)}$, $M' \leftarrow M' \setminus B$
7: **return** P

Example 7 The execution of MPS for the following instance of MTRA is shown in Fig. 2.

Agent	Preferences	Agent	P			
			$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$
1	$1_F 1_B \succ_1 1_F 2_B \succ_1 2_F 2_B \succ_1 2_F 1_B$	1	0.5	0	0	0.5
2	$1_F 2_B \succ_2 2_F 1_B \succ_2 1_F 1_B \succ_2 2_F 2_B$	2	0	0.5	0.5	0

At round 1, agent 1's top bundle is $1_F 1_B$ and agent 2's top bundle is $1_F 2_B$. Notice that both agents wish to consume 1_F. Therefore, round 1 ends as 1_F gets exhausted with both agents getting a share of 0.5 units of 1_F. Agents 1 and 2 also consume 1_B and 2_B, respectively at the same rate during round 1. At the end of round 1, agents 1 and 2 are assigned 0.5 units of $1_F 1_B$ and $1_F 2_B$, respectively.

At the beginning of round 2, there is a supply of 1 unit of 2_F and 0.5 units each of 1_B and 2_B. Agent 1's top available bundle is $2_F 2_B$ since $1_F 2_B$ is unavailable for the exhausted item 1_F, and agent 2's top available bundle is $2_F 1_B$ accordingly. The agents consume the items of each type from their top bundles at a uniform rate. At the end of the round, all items are exhausted, and agents 1 and 2 have consumed 0.5 units each of $2_F 2_B$ and $2_F 1_B$, respectively. This results in the final assignment as shown in Fig. 2.

Note that this output is the undecomposable assignment P in Example 1. Further, we show in Remark 4 that even under lexicographic preferences, the output of MPS is not always decomposable. This means that MPS is only applicable to MTRAs with divisible items. □

Remark 4 The output of MPS is not always a decomposable assignment under the restriction of lexicographic preferences. For the MTRA in Example 4, MPS outputs the following fractional assignment, denoted by P:

Agent	P						
	$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$	$2_F 3_B$	$3_F 2_B$	$3_F 3_B$
1	1/3	0	1/6	1/6	0	1/12	1/4
2	1/3	0	1/6	0	1/6	0	1/3
3	0	1/3	0	1/3	0	1/12	1/4

When items are indivisible, if agent 2 gets $2_F 3_B$, then agent 1 gets $1_F 1_B$ and agent 3 gets $3_F 2_B$ as P indicates. However, $p_{1,1_F 1_B} = 1/3$, $p_{2,2_F 3_B} = 1/6$, and $p_{3,3_F 2_B} = 1/12$ are not equal, a contradiction. □

5.2 Properties of MPS

Under the unrestricted domain of strict linear preferences, Theorem 2 in Wang et al. [42] implies that MPS satisfies sd-efficiency. We prove in Theorem 4 below that MPS satisfies lexi-efficiency, which is a stronger notion of efficiency than sd-efficiency, as we show in Proposition 2.

Theorem 4 *MPS satisfies lexi-efficiency for MTRAs with strict linear preferences.*

Proof Given an MTRA (N, M) and preference profile R, let $P = \text{MPS}(R)$, and suppose that there is another assignment Q satisfying $Q \succ^{lexi} P$. Let N' be the set of agents which have different allocations in Q. By our assumption on Q and strict preferences, for any agent $j \in N'$, there exists a bundle $\mathbf{y}^j$ such that $q_{j,\mathbf{y}^j} > p_{j,\mathbf{y}^j}$ and for every $\mathbf{x} \succ_j \mathbf{y}^j$, $q_{j,\mathbf{x}} = p_{j,\mathbf{x}}$. Let $t_j = \sum_{\mathbf{x} \succ_j \mathbf{y}^j} p_{j,\mathbf{x}}$ be the time agent j takes to consume bundles strictly preferred to $\mathbf{y}^j$. Now, consider the agent $k = \arg\min_{j \in N'} t_j$ with the smallest such time $t_k = \sum_{\mathbf{x} \succ_k \mathbf{y}^k} p_{k,\mathbf{x}}$. However, when MPS executes till time t_k, $\mathbf{y}^k$ is unavailable, which means that at least one item in $\mathbf{y}^k$ is exhausted and $p_{k,\mathbf{y}^k}$ cannot be increased anymore. Therefore, if agent k gains a greater share of $\mathbf{y}^k$ in Q, there is another agent who loses the shares obtained before t_k, which is a contradiction and completes the proof. □

Now, we establish a relationship between lexi-efficiency and sd-efficiency in Proposition 2 through the *no-generalized-cycle* condition (Definition 9), by showing that sd-efficiency is implied by the no-generalized-cycle condition which is equivalent to lexi-efficiency. We begin by borrowing the tool of *generalized cycles* from [42], which is based on the relation τ and the notion of *improvable tuples* defined below.

Definition 8 *(improvable tuples* [42]*)* Given a fractional assignment P and a profile $R = (\succ_j)_{j \le n}$, we define τ as a relation for bundles such that for any $\mathbf{x}, \mathbf{y} \in \mathscr{D}$, $\mathbf{x}\tau\mathbf{y}$ if $\mathbf{x} \succ_j \mathbf{y}$ and $p_{j,\mathbf{y}} > 0$ for some agent $j \le n$. If $\mathbf{x}\tau\mathbf{y}$, then we say that $(\mathbf{x}, \mathbf{y})$ is an *improvable tuple*. $Imp(P, R)$ is the set of all the improvable tuples admitted by assignment P w.r.t. the preference profile R.

For ease of exposition, we use $Imp(P)$ to refer to the set of all the improvable tuples admitted by the fractional assignment P when the profile is clear from the context. We are now ready to formally introduce the no-generalized-cycle condition.

Definition 9 *(no-generalized-cycle* [42]*)* Given an MTRA (N, M) with preference profile R and a fractional assignment P, a set $C \subseteq Imp(P, R)$ is called a *generalized cycle* if for every improvable tuple $(\mathbf{x}^1, \mathbf{y}^1) \in C$, where $\mathbf{x}^1, \mathbf{y}^1 \in \mathscr{D}$, it holds that for every item $o \in \mathbf{x}^1$, there exists an improvable tuple $(\mathbf{x}^2, \mathbf{y}^2) \in C$, where $\mathbf{x}^2, \mathbf{y}^2 \in \mathscr{D}$, such that $o \in \mathbf{y}^2$. We say that P satisfies the *no-generalized-cycle* condition, if it admits no generalized cycles.

When $d = 1$, Bogomolnaia and Moulin [9] proved that an assignment is sd-efficient if and only if the relation τ on it is acyclic, i.e. there does not exist $\mathbf{x}^1\tau\mathbf{x}^2\tau \cdots \tau\mathbf{x}^1$ for $\mathbf{x}^1, \mathbf{x}^2$ and other bundles in $\mathscr{D}$. However, this condition fails for MTRAs. Example 8 shows that an assignment which is not sd-efficient satisfies the acyclicity of τ, but admits a generalized cycle. This suggests that the generalized cycle is more reliable in identifying sd-efficient assignments.

Example 8 We illustrate generalized cycles with the following assignment Q for the MTRA in Example 7. Note that Q is not sd-efficient because the assignment P in Example 7 stochastically dominates Q.

Agent	Q				Agent	Improvable Tuples
	1_F1_B	1_F2_B	2_F1_B	2_F2_B		
1	0.4	0	0	0.6	1	$(1_F1_B, 2_F2_B), (1_F2_B, 2_F2_B)$
2	0.2	0.4	0.4	0	2	$(1_F2_B, 2_F1_B), (1_F2_B, 1_F1_B),$ $(2_F1_B, 1_F1_B)$

It is easy to see that τ is acyclic on Q. However, there is a generalized cycle on Q: $\{(1_F1_B, 2_F2_B), (1_F2_B, 2_F1_B), (2_F1_B, 1_F1_B)\}$. We illustrate this further in Fig. 3 where there is a row for each improvable tuples, and the "Left" and "Right" columns contain the bundles which appear as the left and right component of the improvable tuples respectively. A solid outgoing edge from a bundle $\mathbf{x}$ in the "Left" column to a bundle $\mathbf{y}$ in the "Right" column is used to represent the case where an item of type F in $\mathbf{x}$ is contained in $\mathbf{y}$. Similarly, a dotted edge is used to represent the case where an item of type B in bundle $\mathbf{x}$ in the "Left" column is contained in bundle $\mathbf{y}$ in the "Right" column. We note that such an edge is not unique because one item in the left component may be contained in the right components of several tuples. We do not present all such possible edges for the sake of simplicity and

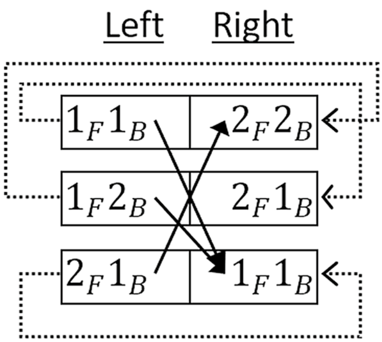


Fig. 3 A generalized cycle for Q in Example 8

clarity. We also note that such a cycle is not unique. Consider for example the items of type B: the item 1_B in the bundle 2_F1_B which is the left component of $(2_F1_B, 1_F1_B)$ is present in 1_F1_B which is the right component of the same tuple, and 2_B in the bundle 1_F2_B which is the left component of $(1_F2_B, 2_F1_B)$ is present in 2_F2_B which is the right component of $(1_F1_B, 2_F2_B)$. A similar correspondence can also be found for each item of type F. □

The proposition below reveals a relationship between lexi-efficiency and sd-efficiency vis-à-vis the no-generalized-cycle condition. Unlike Bogomolnaia [7] who pointed out that lexi-efficiency and sd-efficiency are equivalent in their setting with $d = 1$, we show that this is no longer true for MTRAs. Proposition 2 shows that the no-generalized-cycle condition is equivalent to lexi-efficiency, and they both imply sd-efficiency. After the proposition, we also provide Remark 5 which shows that sd-efficiency does not imply the no-generalized-cycle condition. It also means that sd-efficiency does not imply lexi-efficiency.

Proposition 2 *Given a preference profile R and a fractional assignment P*,

(1) *P is sd-efficient w.r.t. R if P admits no generalized cycle.*
(2) *P is lexi-efficient w.r.t. R if and only if P admits no generalized cycle.*

Proof (1) The idea of proof is similar to the proof of Theorem 5, Claim (1) in [42]. A full proof is provided in "Proof of Proposition 2 (1)" in Appendix for completeness.

(2) *Sufficiency* Suppose by contradiction that P admits no generalized cycle but there exists an assignment $Q \succ^{lexi} P$. Let $N' \subseteq N$ be the set of agents $\{j \in N | Q \succ_j^{lexi} P\}$. For $j \in N \setminus N'$, we have $Q_j = P_j$. For each agent $j \in N'$, let $\mathbf{x}^j$ be the bundle such that $q_{j,\mathbf{x}^j} > p_{j,\mathbf{x}^j}$ and $q_{j,\mathbf{x}} = p_{j,\mathbf{x}}$ for $\mathbf{x} \succ_j \mathbf{x}^j$. For each $\mathbf{x}^j$, there must exist $\mathbf{y}$ such that $\mathbf{x}^j \succ_j \mathbf{y}$ and $p_{j,\mathbf{y}} > 0$. Otherwise by construction we have $\sum_{\mathbf{x}\in\mathcal{D}} q_{j,\mathbf{x}} \geq q_{j,\mathbf{x}^j} + \sum_{\mathbf{x}\succ\mathbf{x}^j} q_{j,\mathbf{x}} = q_{j,\mathbf{x}^j} + \sum_{\mathbf{x}\succ\mathbf{x}^j} p_{j,\mathbf{x}} > p_{j,\mathbf{x}^j} + \sum_{\mathbf{x}\succ\mathbf{x}^j} p_{j,\mathbf{x}} = p_{j,\mathbf{x}^j} + \sum_{\mathbf{x}\succ\mathbf{x}^j} p_{j,\mathbf{x}} + \sum_{\mathbf{x}^j\succ\mathbf{x}} p_{j,\mathbf{x}} = 1$, a contradiction. Then we can build $C_0 = \{(\mathbf{x}^j, \mathbf{y}) | j \in N', \mathbf{x}^j \succ_j \mathbf{x}, p_{j,\mathbf{y}} > 0\}$. For convenience, we define $\mathcal{D}_{L,S}$ and $\mathcal{D}_{R,S}$ to be the set of left and right components in improvable tuple set S, respectively. By construction, we know $\mathcal{D}_{L,C_0} = \{\mathbf{x}^j | j \in N'\}$. The set C_0 is not necessarily a generalized cycle since there may exist items in some bundles $\mathbf{x}^j \in \mathcal{D}_{L,C_0}$ which are not in any bundle in $\mathcal{D}_{R,C_0}$.

To build a generalized cycle, first we provide the following claim, and its proof is provided in "Proof of Claim 3 of Proposition 2" in Appendix.

Claim 3 For any $j \in N'$ and $o \in \mathbf{x}^j$, there exist an agent $k \in N'$ and a bundle $\mathbf{y}$ such that $\mathbf{x}^k \succ_k \mathbf{y}$, $o \in \mathbf{y}$, and $p_{k,\mathbf{y}} > 0$.

Let j be an arbitrary agent in N'. For any $o \in \mathbf{x}^j$, by Claim 3, we can find an agent $k \in N'$, and a bundle $\mathbf{y}$ such that $\mathbf{x}^k \succ_k \mathbf{y}$, $o \in \mathbf{y}$ and $p_{k,\mathbf{y}} > 0$. Note that $(\mathbf{x}^k, \mathbf{y})$ is an improvable tuple. Let $C_j = \{(\mathbf{x}^k, \mathbf{y}) | k \in N'$ and there exists item $o \in \mathbf{x}^j$ such that $o \in \mathbf{y}, \mathbf{x}^k \succ \mathbf{y}, p_{k,\mathbf{y}} > 0\}$. Then, it is easy to see that $\mathscr{D}_{L,C_j} \subseteq \mathscr{D}_{L,C_0}$, and any item $o \in \mathbf{x}^j$ must exist in some bundles in $\mathscr{D}_{R,C_j}$.

Let $C = \bigcup_{j \in N'} C_j \cup C_0$. Then we have that $\mathscr{D}_{L,C} = \mathscr{D}_{L,C_0}$, and for any agent $j \in N'$ and any item o in $\mathbf{x}^j \in \mathscr{D}_{L,C_j} \subseteq \mathscr{D}_{L,C}$, item o is also in some bundles in $\mathscr{D}_{R,C_j} \subseteq \mathscr{D}_{R,C}$. Then, by Definition 9, C is a generalized cycle, which is a contradiction to the assumption that P admits no generalized cycle.

Necessity Suppose for the sake of contradiction that there is a lexi-efficient assignment P which admits a generalized cycle C. We say that an agent j is involved in the tuple $(\mathbf{x}, \mathbf{y})$ if $\mathbf{x} \succ_j \mathbf{y}$ and $p_{j,\mathbf{y}} > 0$. Let $N_C \subseteq N$ be the set of agents who are involved in the tuples in C. The proof involves constructing an assignment Q by applying Steps 1–4 below for each agent j. For an arbitrary agent $j \in N_C$, let $(\mathbf{x}, \mathbf{y}) \in C$ be one of the tuples in which she is involved, and w.l.o.g. let $\mathbf{x}$ be top ranked bundle according to $\succ_j$ among all the bundles in tuples involving agent j. We apply the following steps for agent j:

Step 1 For each item $o_i \in \mathbf{x}$ of type i, we can find an agent $k \in N_C$ which satisfies that there exists a tuple $(\mathbf{x}^k, \mathbf{y}^k) \in C$ such that $\mathbf{x}^k, \mathbf{y}^k \in \mathscr{D}$, $o \in \mathbf{y}^k$ and $p_{k,\mathbf{y}^k} > 0$ by the definition of generalized cycle. For each agent k, we take out her share of $\mathbf{y}^k$ by a small enough value ϵ.

Step 2 We make ϵ units of $\mathbf{x}$ by only extracting the share of each o_i from each $\mathbf{y}^k$ in Step 1 such that $D_{\mathbf{x}}(i) = o_i$ for each $i \leq d$, and we allocate the share of $\mathbf{x}$ to agent j.

Step 3 To keep the supply of bundle not beyond agent j's demand, agent j should give out ϵ units of $\mathbf{y}$.

Step 4 We make ϵ units of $\mathbf{z}^k \in \mathscr{D}$ by combining the share of $\mathbf{y}^k$ without o_i and the share of item $D_i(\mathbf{y})$, i.e. $D_i(\mathbf{z}^k) = D_i(\mathbf{y})$ and $D_h(\mathbf{z}^k) = D_h(\mathbf{y}^k)$ for any $h \neq i$, and we allocate the share of ϵ units of $\mathbf{z}^k$ to agent k.

Let Q be the new assignment after we take the steps above for every agent $j \in N_C$. We note that Q_j exactly meets the demand of agent j for any $j \in N$, and ϵ is chosen to be small enough so that the shares of bundles above are not used up, and therefore they can be reused for other agents. We also note that given the bundle $\mathbf{w}$ which is top ranked according to $\succ_k$ among all the bundles in tuples involving k, we have that $\mathbf{w} \succ_k \mathbf{y}^k$, because agent k is involved in $(\mathbf{x}^k, \mathbf{y}^k) \in C$ and by the selection of $\mathbf{w}$ we have that $\mathbf{x}^k \succ_k \mathbf{y}^k$, and $\mathbf{w} = \mathbf{x}^k$ or $\mathbf{w} \succ_k \mathbf{x}^k$. Therefore, when we take these steps for agent $j \in N_C$, agent k's shares over $U(\mathbf{w}, \succ_k)$ do not decrease. In this way, for any agent j with the selected tuple $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ is top ranked among bundles in all the tuples in which she is involved, we have that: (1) After taking these steps for agent j with the selected tuple $(\mathbf{x}, \mathbf{y})$, agent j obtains ϵ units of $\mathbf{x}$ and loses ϵ units of $\mathbf{y}$. (2) After taking these steps for other agents in N_C where agent j donates some bundles just like agent k in Step 1, agent j loses shares of some bundles in $\{\mathbf{z} | \mathbf{x} \succ_j \mathbf{z}\}$ and obtains shares of some bundles which we do not care about. It follows that $Q_{j,\mathbf{x}} \geq P_{j,\mathbf{x}} + \epsilon > P_{j,\mathbf{x}}$ and for any $j \in N_C$ and $\mathbf{z} \succ_j \mathbf{x}$, we have that $Q_{j,\mathbf{z}} \geq P_{j,\mathbf{z}}$, because agent j does not lose but may gain the shares of these bundles, and therefore $Q \succ_j^{lexi} P$. We also have that $Q_k = P_k$ for $k \in N \setminus N_C$ because agents not in N_C do not take part in the share transferring. Together they imply that $Q \succ^{lexi} P$, which is a contradiction to the assumption that P is lexi-efficient. □

Remark 5 The no-generalized-cycle condition is not a necessary condition of sd-efficiency. Consider an MTRA with two agents where $\succ_1$ and $\succ_2$ are the same as $1_F 1_B \succ_1 1_F 2_B \succ_1 2_F 1_B \succ_1 2_F 2_B$. Consider the following sd-efficient assignment P.

Agent	P			
	$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$
1	0.5	0	0	0.5
2	0	0.5	0.5	0

We see that P admits a generalized cycle $\{(1_F 1_B, 1_F 2_B), (1_F 2_B, 2_F 1_B), (2_F 1_B, 2_F 2_B)\}$. □

Theorem 5 below characterizes the set of all lexi-efficient assignments by the family of *eating algorithms* for MTRAs (Algorithm 3), which is a natural extension of the family of eating algorithms introduced by Bogomolnaia and Moulin [9] for the single type setting. Each eating algorithm is specified by a collection of exogenous *eating speed functions* $\omega = (\omega_j)_{j \leq n}$. An eating speed function ω_j specifies the instantaneous rate at which agent j consumes bundles at each instant $t \in [0, 1]$ such that the integral $\int_{t=0}^{1} \omega_j(t)$ is 1. In each round of an eating algorithm, each agent j consumes her most preferred available bundle at the rate specified by her eating speed function ω_j, until the supply of one of the items in one of the bundles being consumed is exhausted. Note that MPS is a special case of the family of eating algorithms, with $\omega_j(t) = 1$ for any $t \in [0, 1]$ and $j \in N$.

Algorithm 3 Eating Algorithms

1: **Input:** An MTRA (N, M) and a preference profile R.
2: **Parameters:** Eating speed functions $\omega = (\omega_j)_{j \leq n}$.
3: For each $o \in M$, $supply(o) \leftarrow 1$. $M' \leftarrow M$. $P \leftarrow 0^{n \times |\mathcal{D}|}$. $t \leftarrow 0$.
4: **while** $M' \neq \emptyset$ and $t < 1$ **do**
5: **Identify top bundle** $top(j)$ for each agent $j \leq n$.
6: **Consume.**
5.1: For each $o \in M'$, $consumers(o) \leftarrow |\{j \in N | o \in top(j)\}|$.
5.2: $\rho \leftarrow \min\{\rho | \sum_{j \in consumers(o)} \int_t^{t+\rho} \omega_j = supply(o), o \in M'\}$.
5.3: For each $j \leq n$, $p_{j,top(j)} \leftarrow p_{j,top(j)} + \int_t^{t+\rho} \omega_j$.
5.4: For each $o \in M'$, $supply(o) \leftarrow supply(o) - \sum_{j \in consumers(o)} \int_t^{t+\rho} \omega_j$.
7: $M' \leftarrow M' \setminus \{o \in M' | supply(o) = 0\}$. $t \leftarrow t + \rho$.
8: **return** P

Theorem 5 *Given an MTRA, an assignment is lexi-efficient if and only if it is the output of an eating algorithm (Algorithm 3).*

By Proposition 2, we see that the lexi-efficient assignments are also the ones satisfying the no-generalized-cycle condition. Therefore, we consider the assignments satisfying the no-generalized-cycle condition instead in the proof of Theorem 5 which is provided in "Proof of Theorem 5" in Appendix.

For MTRAs with CP-net preferences, Theorem 5 in [42] showed that MPS satisfies sd-envy-freeness. Here CP-net determines the dependence among preferences of types, which also reflects the importance of each type. Since the domain of CP-net preferences and strict linear preferences are not totally overlapping, we provide Proposition 3 as a complement and the proof of the proposition is in "Proof of Proposition 3" in Appendix.

Proposition 3 *MPS satisfies sd-envy-freeness for MTRAs with strict linear preferences.*

As we show in Remark 6, MPS does not satisfy sd-weak-strategyproofness. Fortunately, as we prove in Theorem 6, MPS does satisfy sd-weak-strategyproofness under lexicographic preferences. Importantly, this is true even when agents may have different importance orders. This is in contrast with the result in Wang et al. [42], who show that under the domain of CP-net preferences, MPS satisfies sd-weak-strategyproof only if all agents' preferences share a trivial dependency structure where all the types are independent, meaning that all the types are of equal importance.

Remark 6 MPS does not satisfy sd-weak-strategyproofness for MTRAs with strict linear preferences. Consider an MTRA with two agents where $\succ_1$ and $\succ_2$ are in the following:

Agent	Preferences
1	$1_F2_B \succ_1 1_F1_B \succ_1 2_F1_B \succ_1 2_F2_B$
2	$1_F1_B \succ_2 2_F1_B \succ_2 2_F2_B \succ_2 1_F2_B$

MPS outputs P for this preference profile. If agent 1 misreports $\succ_1$ as $\succ_1'$: $2_F1_B \succ_1' 1_F1_B \succ_1' 1_F2_B \succ_1' 2_F2_B$, then MPS outputs P'. Both P and P' are shown as below:

Agent	P				Agent	P'			
	1_F1_B	1_F2_B	2_F1_B	2_F2_B		1_F1_B	1_F2_B	2_F1_B	2_F2_B
1	0	0.5	0.25	0.25	1	0	0.5	0.5	0
2	0.5	0	0.25	0.25	2	0.5	0	0	0.5

We have the fact that $P' \succeq_1^{sd} P$ and $P' \neq P$, which does not satisfy the requirement of sd-weak-strategyproofness i.e. $P_1' = P_1$ if $P' \succeq_1^{sd} P$. □

Theorem 6 *MPS satisfies sd-weak-strategyproofness for MTRAs with lexicographic preferences.*

Proof Consider an arbitrary MTRA (N, M) and an arbitrary lexicographic preference profile R. Suppose for the sake of contradiction that an agent j can obtain a better allocation by misreporting her preference as another lexicographic preference $\succ_j'$. Throughout, we set $P = \text{MPS}(R)$ and $Q = \text{MPS}(R')$, where $R' = (\succ_j', \succ_{-j})$. W.l.o.g. let the types be labeled such that $1 \rhd_j \cdots \rhd_j d$. By the assumption of beneficial misreporting, we have $Q \succeq_j^{sd} P$ and we need to prove that $Q_j = P_j$.

We show that $Q_j^i = P_j^i$ for any type $i \leq d$ where Q_j^i and P_j^i are agent j's single type allocations of type i in Q and P, respectively. In fact, we only need to show $Q^i \succeq_j^{sd} P^i$ due to following claim:

Claim 4 Under lexicographic preferences, $(\text{MPS}(R))^i = PS(R^i)$, where $R^i = (\succ_j^i)_{j \leq n}$.

The claim is obtained by comparing the execution of MPS with PS in each type. The full proof of the claim is in "Proof of Claim 4 in Theorem 6" in Appendix. Since PS satisfies sd-weak-strategyproofness [9], we deduce from Claim 4 that $Q_j^i = P_j^i$ if $Q^i \succeq_j^{sd} P^i$ for each type i. Therefore, we can prove $Q_j^i = P_j^i$ by showing $Q^i \succeq_j^{sd} P^i$ instead. We prove it by mathematical induction on type i.

In the following discussion, we recall the notations used in Sect. 4.2. For the (partial) bundle $\mathbf{w} \in \prod_{i \in H} D_i$ where $H \subseteq \{i | i \leq d\}$, we use $w_i = D_i(\mathbf{w})$ and $Z_{\mathbf{w}} = \{\mathbf{x} \in \mathscr{D} | \text{for all } i \in H, x_i = w_i\}$. We also use $(o_i)_{i \in H}$ to refer to an bundle containing items o_i of each type $i \in H$.

Base case First, we prove that $Q^1 \succeq_j^{sd} P^1$. Assume $Q^1 \not\succeq_j^{sd} P^1$. It means that there exists y_1 such that $\sum_{o \in U(\succ_j^i, y_1)} p_{j,o}^1 > \sum_{o \in U(\succ_j^i, y_1)} q_{j,o}^1$. Let $\mathbf{y}$ be the least preferred bundle containing y_1. It follows that

$$\sum_{o \in U(\succ_j^i, y_1)} p_{j,o}^1 = \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} > \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} q_{j,\mathbf{x}} = \sum_{o \in U(\succ_j^i, y_1)} q_{j,o}^1,$$

which is a contradiction to our assumption. Thus $Q^i \succeq_j^{sd} P^i$, i.e. $Q_j^1 = P_j^1$.

Inductive step Next, consider any type i such that $1 < i \leq d$ and suppose that for every $h < i$, it holds that $Q_j^h = P_j^h$. We show that $Q^i \succeq_j^{sd} P^i$. Let $\mathbf{y}$ be the bundle having y_h as an item of type h for any $h \leq i$. W.l.o.g. let $\mathbf{y}$ be the least preferred bundle in $Z_{(y_h)_{h \leq i}}$. Let $S_i = \{\mathbf{x} \in Z_{(y_h)_{h \leq i-1}} | x_i \in U(\succ_j^i, y_i)\}$. Because $Q \succeq_j^{sd} P$, we have that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} \leq \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} q_{j,\mathbf{x}}$. We split the shares over the upper contour set as follows:

$$\begin{aligned} \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} = & \sum_{\mathbf{x} \in \{\mathbf{x} | x_1 \succ_j^1 y_1\}} p_{j,\mathbf{x}} + \sum_{\mathbf{x} \in \{\mathbf{x} \in Z_{(y_1)} | x_2 \succ_j^2 y_2\}} p_{j,\mathbf{x}} + \dots \\ & + \sum_{\mathbf{x} \in \{\mathbf{x} \in Z_{(y_h)_{h \leq i-2}} | x_{i-1} \succ_j^{i-1} y_{i-1}\}} p_{j,\mathbf{x}} + \sum_{\mathbf{x} \in S_i} p_{j,\mathbf{x}} \end{aligned} \tag{16}$$

Claim 5 below is obtained from the observation that agents consume bundles until they are unavailable in MPS. The full proof of the claim is in "Proof of Claim 5 in Theorem 6" in Appendix.

Claim 5 For $P = \text{MPS}(R)$ and type i, if Q satisfies that $Q_j^h = P_j^h$ for any $h \leq i$ and $Q \succeq_j^{sd} P$, then $\sum_{\mathbf{x} \in S_i} p_{j,\mathbf{x}} = \sum_{\mathbf{x} \in S_i} q_{j,\mathbf{x}}$ for any $y_1, \dots, y_{i-1}, y_i$ and $S_i = \{\mathbf{x} \in Z_{(y_h)_{h \leq i-1}} | x_i \in U(\succ_j^i, y_i)\}$.

By our assumption that $Q_j^h = P_j^h$ for any $h < i$, by Claim 5 we have that $\sum_{\mathbf{x} \in S_h} p_{j,\mathbf{x}} = \sum_{\mathbf{x} \in S_h} q_{j,\mathbf{x}}$ for $h < i$, and therefore due to the fact that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}} \leq \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} q_{j,\mathbf{x}}$ and the Eq. (16), we have that $\sum_{\mathbf{x} \in S_i} p_{j,\mathbf{x}} \leq \sum_{\mathbf{x} \in S_i} q_{j,\mathbf{x}}$. By summing up each side over all the possible choices of $y_1, \dots, y_{i-1}$, we have that

$$\sum_{y_1}\sum_{y_2}\cdots\sum_{y_{i-1}}\sum_{\mathbf{x}\in S_i} p_{j,\mathbf{x}} \leq \sum_{y_1}\sum_{y_2}\cdots\sum_{y_{i-1}}\sum_{\mathbf{x}\in S_i} q_{j,\mathbf{x}}.$$

Then, after simplifying we have that

$$\sum_{\mathbf{x}\in\{\mathbf{x}|x_i\in U(\succ_j^i,y_i)\}} p_{j,\mathbf{x}} \leq \sum_{\mathbf{x}\in\{\mathbf{x}|x_i\in U(\succ_j^i,y_i)\}} q_{j,\mathbf{x}}.$$

It means that $\sum_{o\in U(\succ_j^i,y_i)} p_{j,o}^i \leq \sum_{o\in U(\succ_j^i,y_i)} q_{j,o}^i$. This also implies that $Q^i \succeq_j^{sd} P^i$, i.e. $Q_j^i = P_j^i$.

This proves that $Q_j^i = P_j^i$ for any $i \leq d$ by induction. By Claim 5 for type $i = d$, we have that $\sum_{\mathbf{x}\in S_d} p_{j,\mathbf{x}} = \sum_{\mathbf{x}\in S_d} q_{j,\mathbf{x}}$. It also means that $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$, i.e. $Q_j = P_j$ which completes the proof. □

5.3 Characterizations of MPS

In Theorem 7 we provide two characterizations of MPS. Before we show the theorem, we introduce the two properties involved in the characterizations. Leximin-optimality requires that the assignment leximin maximizes the vector describing cumulative shares at each bundle [5, 7, 10], which reflects the egalitarian nature of the mechanism in attempting to equalize agents' shares of their top ranked choices. The definition uses the following notation: for any vector $\mathbf{u}$ of length k, $\mathbf{u}^* = (u_1^*, u_2^*, \dots, u_k^*)$ is its transformation into the k-vector of $\mathbf{u}$'s components sorted in ascending order.

Definition 10 *(leximin-optimality)* Let L be the *leximin relation*, where for any two vectors $\mathbf{u}, \mathbf{v}$, we say that $(\mathbf{u}, \mathbf{v}) \in L$ if there exists k such that $\mathbf{u}_k^* > \mathbf{v}_k^*$ and $\mathbf{u}_l^* = \mathbf{v}_l^*$ for $l < k$. For any fractional assignment P, let $\mathbf{u}^P = (u_{j,\mathbf{x}}^P)_{j\leq n,\mathbf{x}\in\mathscr{D}}$, where $u_{j,\mathbf{y}}^P = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}}$ for each agent $j \leq n$ and bundle $\mathbf{y} \in \mathscr{D}$. A fractional assignment P is *leximin-optimal*, if $(\mathbf{u}^P, \mathbf{u}^Q) \in L$ for any other assignment $Q \in \mathscr{P}$. A mechanism f satisfies leximin-optimality if $f(R)$ is leximin-optimal for any $R \in \mathscr{R}$.

Example 9 For the MTRA and the two assignments Q and Q' in Example 2, the elements of $\mathbf{u}^Q$ and $\mathbf{u}^{Q'}$ are listed in the following table:

Agent	$\mathbf{u}^Q$				Agent	$\mathbf{u}^{Q'}$			
	$u^Q_{1_F1_B}$	$u^Q_{1_F2_B}$	$u^Q_{2_F1_B}$	$u^Q_{2_F2_B}$		$u^{Q'}_{1_F1_B}$	$u^{Q'}_{1_F2_B}$	$u^{Q'}_{2_F1_B}$	$u^{Q'}_{2_F2_B}$
1	0.25	0.5	0.75	1	1	0.25	0.75	0.75	1
2	0.25	0.75	0.5	1	2	0.25	0.75	0.75	1

We use $\mathbf{u} = \mathbf{u}^Q$ and $\mathbf{v} = \mathbf{u}^{Q'}$ for short. We rearrange them in the ascending order and have that $\mathbf{u}^* = (0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1)$ and $\mathbf{v}^* = (0.25, 0.25, 0.75, 0.75, 0.75, 0.75, 1, 1)$. Then we have that $v_1^* = u_1^*$, $v_2^* = u_2^*$, and $v_3^* > u_3^*$, which means that $(\mathbf{u}^{Q'}, \mathbf{u}^Q) \in L$ by definition. □

Hashimoto et al. [24] provided a characterization of PS with a single property named *ordinal fairness* which involves the comparison of the cumulative shares of items. We extend *ordinal fairness* to MTRAs as item-wise ordinal fairness and provide a similar characterization of MPS. In contrast to sd-envy-freeness, the upper contour sets in item-wise ordinal fairness depend on the different preferences, and the bundles to determine the sets only need to share a certain item. We note that item-wise ordinal fairness involves the cumulative shares over bundles containing a certain item, different from the version in [42] which involves the share of each bundle.

Definition 11 *(item-wise ordinal fairness)* A fractional assignment P is item-wise ordinal fair if P satisfies the condition that for any agent j and bundle $\mathbf{y}$ with $p_{j,\mathbf{y}} > 0$, there exists an item $o \in \mathbf{y}$ such that $\sum_{\mathbf{x}\in U(\succ_k,\mathbf{z})} p_{k,\mathbf{x}} \leq \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}}$ for any agent k and bundle $\mathbf{z}$ with $o \in \mathbf{z}$ and $p_{k,\mathbf{z}} > 0$. A mechanism f satisfies item-wise ordinal fairness if $f(R)$ is item-wise ordinal fair for any $R \in \mathscr{R}$.

Example 10 We revisit the MTRA in Example 2 and show that the assignment Q' in it is not item-wise ordinal fair. In the following table, we list the share of each bundle and the accumulated share at that bundle for each agent in Q'. We note that the order of bundles in the table are rearranged according to each agent's preference.

	Shares			
$\mathbf{x}$	1_F1_B	1_F2_B	2_F1_B	2_F2_B
$q'_{1,\mathbf{x}}$	0.25	0.5	0	0.25
$\sum_{\mathbf{y}\in U(\succ_1,\mathbf{x})} q'_{1,\mathbf{y}}$	0.25	0.75	0.75	1
$\mathbf{x}$	1_F1_B	2_F1_B	1_F2_B	2_F2_B
$q'_{2,\mathbf{x}}$	0.25	0.5	0	0.25
$\sum_{\mathbf{y}\in U(\succ_2,\mathbf{x})} q'_{2,\mathbf{y}}$	0.25	0.75	0.75	1

We use 1_F1_B of which agent 1 has positive shares as an example to show that no item o in 1_F1_B satisfying that $q'_{j_\mathbf{y}} > 0$ and $\sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} q'_{j,\mathbf{x}} \leq \sum_{\mathbf{x}\in U(\succ_1,1_F1_B)} q'_{1,\mathbf{x}}$ for any agent j and $\mathbf{y}$ containing o. For item 1_F in 1_F1_B, we can find out the bundle 1_F2_B containing 1_F and $q'_{1,1_F2_B} = 0.5 > 0$ for agent 1, but $\sum_{\mathbf{x}\in U(\succ_1,1_F2_B)} q'_{1,\mathbf{x}} = 0.75 > 0.25 = \sum_{\mathbf{x}\in U(\succ_1,1_F1_B)} q'_{1,\mathbf{x}}$. Similarly for item 1_B in 1_F1_B, we can find out the bundle 2_F1_B containing 1_B and $q'_{2,2_F1_B} = 0.5 > 0$ for agent 2, but $\sum_{\mathbf{x}\in U(\succ_2,2_F1_B)} q'_{2,\mathbf{x}} = 0.75 > 0.25 = \sum_{\mathbf{x}\in U(\succ_1,1_F1_B)} q'_{1,\mathbf{x}}$. □

Theorem 7 *Under the domain of strict linear preferences,*

(I) *MPS is the unique mechanism which satisfies* leximin-optimality, *and*
(II) *MPS is the unique mechanism which satisfies* item-wise ordinal fairness.

Proof *(I) leximin-optimality* Given an MTRA (N, M) with any profile of strict linear preferences R, let $P = \text{MPS}(R)$ and $\mathbf{u} = (u_{j,\mathbf{x}})_{j\leq n,\mathbf{x}\in\mathscr{D}}$. For $j \leq n$ and bundle $\mathbf{x} \in \mathscr{D}$, let $u_{j,\mathbf{x}} = \sum_{\mathbf{y}\in U(\succ_j,\mathbf{x})} p_{j,\mathbf{y}}$. Let Q be an arbitrary fractional assignment which is leximin-optimal w.r.t. the vector $\mathbf{v} = (v_{j,\mathbf{x}})_{j\leq n,\mathbf{x}\in\mathscr{D}}$ where $v_{j,\mathbf{x}} = \sum_{\mathbf{y}\in U(\succ_j,\mathbf{x})} q_{j,\mathbf{y}}$. In the following proof, we show that $\mathbf{v}^* = \mathbf{u}^*$ which means that P is leximin-optimal (satisfaction), and the assignments P and Q are identical which means that P is the unique leximin-optimal assignment (uniqueness). We prove that $v^*_k = u^*_k$ and if $u_{j,\mathbf{x}} = u^*_k$, then $q_{j,\mathbf{x}} = p_{j,\mathbf{x}}$ for any agent j by induction on k.

Base case We prove that $u_1^* = v_1^*$ and if $u_{j,\mathbf{x}} = u_1^*$, then $q_{j,\mathbf{x}} = p_{j,\mathbf{x}}$ for any agent j. By the selection of Q, we know that $u_1^* \leq v_1^*$. Suppose for the sake of contradiction that $u_1^* < v_1^*$. We use the tuple $(j, \mathbf{x})$ as the index of the component $u_{j,\mathbf{x}}$ for any agent $j \leq n$ and bundle $\mathbf{x} \in \mathscr{D}$. Let S_k be the set of indices such that for each $(j, \mathbf{x}) \in S_k$, $u_{j,\mathbf{x}} = u_k^*$.

We consider the corresponding elements of $\mathbf{v}$ indicated by the set S_1. We note that for each $(j, \mathbf{x}) \in S_1$, there are two possible cases:

Case (i) $\mathbf{x}$ is the most preferred bundle w.r.t. $\succ_j$. Then, $p_{j,\mathbf{x}} = \sum_{\mathbf{y}\in U(j,\mathbf{x})} p_{j,\mathbf{y}} = u_1^*$, and $p_{j,\mathbf{x}} = u_1^* \leq v_1^* \leq \sum_{\mathbf{y}\in U(j,\mathbf{x})} q_{j,\mathbf{y}} = q_{j,\mathbf{x}}$. The assumption that $u_1^* < v_1^*$ implies that $p_{j,\mathbf{x}} < q_{j,\mathbf{x}}$.
Case (ii) $\mathbf{x}$ is *not* the most preferred bundle w.r.t. $\succ_j$. Then, for the most preferred bundle $\mathbf{z}$ w.r.t. $\succ_j$, there must exist $p_{j,\mathbf{z}} = u_1^*$ as in *Case 1*, since $u_1^* \leq \sum_{\mathbf{y}\in U(j,\mathbf{z})} p_{j,\mathbf{y}} \leq \sum_{\mathbf{y}\in U(j,\mathbf{x})} p_{j,\mathbf{y}} = u_1^*$. This implies that $p_{j,\mathbf{x}} = 0 \leq q_{j,\mathbf{x}}$.

From the execution of MPS, $\mathbf{x}$ must be unavailable at time u_1^* because some items in it are exhausted at that time. Let B_1 denote the set of the items exhausted at time u_1^*. For any $o \in B_1$, we have that $\sum_{(a,\mathbf{y})\in S_1, o\in\mathbf{y}} p_{a,\mathbf{y}} = 1$. With the inequalities in Cases (i) and (ii) we have that $\sum_{(a,\mathbf{y})\in S_1, o\in\mathbf{y}} q_{a,\mathbf{y}} > \sum_{(a,\mathbf{x})\in S_1, o\in\mathbf{y}} p_{a,\mathbf{y}} = 1$ for some $o \in B_1$, which is a contradiction. Therefore, we have that $u_1^* = v_1^*$ and $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$ for any $(j, \mathbf{x}) \in S_1$, i.e. $u_{j,\mathbf{x}} = u_1^*$. We also have that $u_k^* = u_1^* = v_k^*$ for all $k \leq |S_1|$ trivially.

Inductive step For any $k > 1$ with $u_k^* > u_{k-1}^*$, suppose that $u_l^* = v_l^*$ for any $l < k$ and $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$ for any $(j, \mathbf{x}) \in S_l$ with $l < k$. We prove that $u_k^* = v_k^*$ and $q_{j,\mathbf{x}} = p_{j,\mathbf{x}}$ if $u_{j,\mathbf{x}} = u_k^*$, i.e. $(j, \mathbf{x}) \in S_k$. By the selection of Q, we know that $u_k^* \leq v_k^*$. Suppose for the sake of contradiction that $u_k^* < v_k^*$. For any $(j, \mathbf{x}) \in S_k$, let $\mathbf{y}$ be the least preferred bundle in $\{\mathbf{z}|\mathbf{z} \succ_j \mathbf{x}\}$ w.r.t. $\succ_j$ and its corresponding index is $(j, \mathbf{y})$. Then, we have that $p_{j,\mathbf{x}} = u_k^* - u_{j,\mathbf{y}}$. Let $u_l^* = u_{j,\mathbf{y}}$. By the initial assumption that $(\mathbf{v}, \mathbf{u}) \in L$, we have that $p_{j,\mathbf{x}} \leq q_{j,\mathbf{x}}$ for any $(j, \mathbf{x}) \in S_k$, because

Case (i') if $(j, \mathbf{y}) \notin S_k$, then we have that $l < k$ and $u_k^* > u_l^* = v_l^*$, and therefore $p_{j,\mathbf{x}} = u_k^* - u_l^* \leq v_k^* - v_l^* = q_{j,\mathbf{x}}$. The assumption that $u_k^* < v_k^*$ implies that $p_{j,\mathbf{x}} < q_{j,\mathbf{x}}$.
Case (ii') if $(j, \mathbf{y}) \in S_k$, then $u_l^* = u_k^*$ and $p_{j,\mathbf{x}} = 0 \leq q_{j,\mathbf{x}}$.

W.l.o.g. let $\mathbf{x}$ satisfy $p_{j,\mathbf{x}} < q_{j,\mathbf{x}}$. We know $\mathbf{x}$ is unavailable at time u_k^* in the execution of MPS because of exhausted items in it. Let B_k be the set of items exhausted at time u_k^*. For any $o \in B_k$, we have that

$$\begin{aligned} \sum_{(a,\mathbf{z})\in\cup_{l<k}S_l, o\in\mathbf{z}} p_{a,\mathbf{z}} &= \sum_{(a,\mathbf{z})\in\cup_{l<k}S_l, o\in\mathbf{z}} q_{a,\mathbf{z}}, \\ \sum_{(a,\mathbf{z})\in\cup_{l<k}S_l, o\in\mathbf{z}} p_{a,\mathbf{z}} + \sum_{(a,\mathbf{z})\in S_k, o\in\mathbf{z}} p_{a,\mathbf{z}} &= 1. \end{aligned} \tag{17}$$

Thus we see that $\sum_{(a,\mathbf{z})\in\cup_{l\leq k}S_l, o\in\mathbf{x}} q_{a,\mathbf{z}} > \sum_{(a,\mathbf{z})\in\cup_{l\leq k}S_l, o\in\mathbf{x}} p_{a,\mathbf{z}} = 1$ for some $o \in B_k$, which is a contradiction. Therefore, it follows that $u_k^* = v_k^*$ and $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$ for any $(j, \mathbf{x}) \in S_k$ by Cases (i') and (ii'). We also know that $u_l^* = u_k^* = v_l^*$ for any l with $k \leq l < k + |S_k|$.

By induction, we have that $v_k^* = u_k^*$ for any k, i.e. $\mathbf{u} = \mathbf{v}$, and $q_{j,\mathbf{x}} = p_{j,\mathbf{x}}$ for any $j \in N$ and $\mathbf{x} \in \mathscr{D}$, i.e. $P = Q$. Together we have that P is the unique leximin-optimal assignment for the given MTRA, which means that MPS is the unique mechanism which satisfies leximin-optimality.

(II) item-wise ordinal fairness We use the relationship between time and consumption during the execution of MPS in the proof and to show uniqueness. Given any MTRA (N, M) and the preference profile R, let $P = \text{MPS}(R)$ in the following proof.

Satisfaction For an arbitrary agent j and bundle $\mathbf{x}$, let $t_\mathbf{x} = \sum_{\mathbf{z}\in U(\succ_j,\mathbf{x})} p_{j,\mathbf{z}}$. Assume for the sake of contradiction that for every $o \in \mathbf{x}$, there exists an agent k and a bundle $\mathbf{y}$ such that $o \in \mathbf{y}$, $p_{k,\mathbf{y}} > 0$, and $t_\mathbf{y} = \sum_{\mathbf{z}\in U(\succ_k,\mathbf{y})} p_{k,\mathbf{z}} > t_\mathbf{x}$. From the relation of time and consumption, we know that at time $t_\mathbf{x}$, $\mathbf{x}$ is unavailable and therefore the supply of some item in $\mathbf{x}$ is exhausted.

Now, let us fix $o' \in \mathbf{x}$ to be an item that is exhausted at time $t_\mathbf{x}$ when $\mathbf{x}$ is unavailable. By our assumption, we know that there exist a bundle $\mathbf{w}$ and an agent l such that $o' \in \mathbf{w}$, $p_{l,\mathbf{y}} > 0$, and $t_\mathbf{w} = \sum_{\mathbf{z}\in U(\succ_l,\mathbf{w})} p_{l,\mathbf{z}} > t_\mathbf{x}$. By the assumption, we know that $\mathbf{w}$ is available during $[0, t_\mathbf{w}]$ and $t_\mathbf{w} > t_\mathbf{x}$, which also means that $o' \in \mathbf{w}$ is not exhausted after $t_\mathbf{x}$, a contradiction to o is exhausted at time $t_\mathbf{x}$.

Uniqueness Suppose that $Q \neq P$ is an item-wise ordinal fair assignment for the sake of contradiction. Let $t_{j,\mathbf{x}}$ be the smaller quantity among $\sum_{\mathbf{z}\in U(\succ_j,\mathbf{x})} p_{j,\mathbf{z}}$ and $\sum_{\mathbf{z}\in U(\succ_j,\mathbf{x})} q_{j,\mathbf{z}}$ for any agent j and bundle $\mathbf{x}$. Let t be the smallest among the set $T = \{t_{j,\mathbf{x}} | j \in N, \mathbf{x} \in \mathcal{D}, \sum_{\mathbf{z}\in U(\succ_j,\mathbf{x})} p_{j,\mathbf{z}} \neq \sum_{\mathbf{z}\in U(\succ_j,\mathbf{x})} q_{j,\mathbf{z}}\}$. We note T is not empty because P and Q are different.

Now, w.l.o.g. let agent j and $\mathbf{x}$ satisfy that $t_{j,\mathbf{x}} = t$, and we have that $p_{j,\mathbf{x}} \neq q_{j,\mathbf{x}}$ and $p_{k,\mathbf{y}} = q_{k,\mathbf{y}}$ for $(k, \mathbf{y}) \in S = \{(k, \mathbf{y}) | \sum_{\mathbf{z}\in U(\succ_k,\mathbf{y})} q_{j,\mathbf{z}} \leq t\}$.

We first consider the case that $p_{j,\mathbf{x}} < q_{j,\mathbf{x}}$, i.e. $t = \sum_{\mathbf{z}\in U(\succ_j,\mathbf{x})} p_{j,\mathbf{z}}$, which means that agent j gets a greater share of $\mathbf{x}$ in Q and therefore demands more supply of item contained in $\mathbf{x}$. We also know that in P there exists an item $o \in \mathbf{x}$ exhausted at t which makes $\mathbf{x}$ unavailable. It means that the supply of o is also used up in Q by assigning bundle $\mathbf{y}$ to agent k for some $(k, \mathbf{y}) \in S$. Therefore, the extra demand of o for agent j on bundle $\mathbf{x}$ comes from the share of some bundle $\mathbf{y}$ with $o \in \mathbf{y}$ held by agent k such that $(k, \mathbf{y}) \in S$, which means that $q_{k,\mathbf{y}} < p_{k,\mathbf{y}}$, a contradiction.

Then we consider $p_{j,\mathbf{x}} > q_{j,\mathbf{x}}$ i.e. $t = \sum_{\mathbf{z}\in U(\succ_j,\mathbf{x})} q_{j,\mathbf{z}}$ now, which means that agent j gives up some shares of $\mathbf{x}$ in Q, and thus for any $o \in \mathbf{x}$, we can find some agent k and bundle $\mathbf{y}$ containing o such that k gains a greater share of $\mathbf{y}$ in Q, i.e. $q_{k,\mathbf{y}} > p_{k,\mathbf{y}} \geq 0$. By the selection of j and $\mathbf{x}$, we have that $\sum_{\mathbf{z}\in(\succ_k,\mathbf{y})} q_{j,\mathbf{z}} \geq t = \sum_{\mathbf{z}\in(\succ_j,\mathbf{x})} q_{j,\mathbf{z}}$. We claim that $\sum_{\mathbf{z}\in(\succ_k,\mathbf{y})} q_{j,\mathbf{z}} > t$. Otherwise, if $\sum_{\mathbf{z}\in(\succ_k,\mathbf{y})} q_{j,\mathbf{z}} = t = t_{k,\mathbf{z}}$, then we have that $\sum_{\mathbf{z}\in(\succ_k,\mathbf{w})} q_{j,\mathbf{z}} \leq t$ for $\mathbf{w} \succ_k \mathbf{y}$ i.e. $(k, \mathbf{w}) \in S$, and therefore $p_{k,\mathbf{w}} = q_{k,\mathbf{w}}$. With $q_{k,\mathbf{y}} > p_{k,\mathbf{y}}$, it follows that $\sum_{\mathbf{z}\in(\succ_k,\mathbf{w})} q_{j,\mathbf{z}} > \sum_{\mathbf{z}\in(\succ_k,\mathbf{w})} p_{j,\mathbf{z}}$, which contradicts the selection of $t_{k,\mathbf{w}}$. Therefore, we have that $\sum_{\mathbf{z}\in(\succ_k,\mathbf{y})} q_{j,\mathbf{z}} > \sum_{\mathbf{z}\in(\succ_j,\mathbf{x})} q_{j,\mathbf{z}}$, a contradiction to the assumption that Q is item-wise ordinal fair.

Together, we have that $t = p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$, which contradicts the fact that t is the smallest among the set T and completes the proof. □

6 Conclusion and future work

In this paper, we have showed that it is impossible to design sd-efficient and sd-envy-free mechanisms with decomposable outputs for MTRAs with indivisible items under the unrestricted domain of strict preferences over bundles. Fortunately, under the natural assumption that agents' preferences are lexicographic, this impossibility result is circumvented, as we have showed by proposing the LexiPS mechanism and proving that it is able to deal with indivisible items while satisfying the desirable efficiency and fairness properties of sd-efficiency and sd-envy-freeness.

For divisible items, we have showed that the existing MPS mechanism satisfies the stronger efficiency notion of lexi-efficiency in addition to sd-envy-freeness under the unrestricted domain of linear preferences, and is sd-weak-strategyproof under lexicographic preferences, which complement the results in Wang et al. [42]. In addition, we have provided two characterizations of MPS with leximin-optimality and item-wise ordinal fairness, respectively.

Characterizing the domain of preferences under which it is possible to design mechanisms for MTRAs with indivisible items that are simultaneously fair, efficient, and strategyproof is an exciting topic for future research. Another interesting direction is characterizing mechanisms satisfying sd-efficiency and sd-envy-freeness with other combinations of desirable properties [17] for MTRAs with divisible items. In addition, it is also an exciting avenue for future research to develop efficient and fair mechanisms for natural extensions of the MTRA problem such as settings where there are demands for multiple units of each type, or initial endowments.

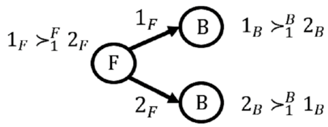
Appendix

Proof of Proposition 1

To prove the tightened impossibility result for MTRAs, we construct a profile R of LP-tree preferences below.

First, we provide $\succ_1$ as an example of LP-tree preferences in the form of a rooted directed tree. Notice that the node labeled by F has a preference $1_F \succ_1^F 2_F$ attached to it and the outgoing edges from this node are labeled by items 1_F and 2_F, respectively. We also note that the types F and B occur once on each branch.

Agent	Preferences
1	$1_F 1_B \succ_1 1_F 2_B \succ_1 2_F 2_B \succ_1 2_F 1_B$
2	$1_F 2_B \succ_2 1_F 1_B \succ_2 2_F 1_B \succ_2 2_F 2_B$



Now, we proceed with the proof. Suppose that f satisfies sd-weak-efficiency and sd-weak-envy-freeness, and let $Q = f(R)$. Suppose for the sake of contradiction that Q is decomposable. Then, it must be possible to represent Q in the form below.

Agent	Q			
	$1_F 1_B$	$1_F 2_B$	$2_F 1_B$	$2_F 2_B$
1	v	w	y	z
2	z	y	w	v

We now show that such an assignment does not exist. We observe that by our assumption of sd-weak-efficiency and agent 1's preference, it is not possible for agent 1 to get shares of 1_F2_B and 2_F1_B simultaneously, i.e. either y or w is 0, or they are both 0. Otherwise, she can improve her allocation by getting shares of 1_F1_B and 2_F2_B which she prefers to 1_F2_B and 2_F1_B by swapping the items 2_B and 1_B in 1_F2_B and 2_F1_B, respectively with agent 2. From agent 2's preferences, we observe that this swap is also preferred by agent 2. By the same token, agent 2 does not get shares of 1_F1_B and 2_F2_B simultaneously, i.e. either v or z is 0, or they are both 0.

Due to $v + w + y + z = 1$, we see that not all of them are 0. Hence, we consider the cases where three of them are 0, which mean that Q is a discrete assignment. The assignments that assign any agent with her least preferable bundle, i.e. 2_F1_B to agent 1 or 2_F2_B to agent 2, are excluded because it violates sd-weak-envy-freeness, which means $y = v = 0$. The other possible cases are: If $w \neq 0$, i.e. Q assigns 1_F2_B to agent 1 and 2_F1_B to agent 2, then agent 2 envies agent 1 due to $1_F2_B \succ_2 2_F1_B$. If $z \neq 0$, i.e. Q assigns 2_F2_B to agent 1 and 1_F1_B to agent 2, then agent 1 envies agent 2 due to $1_F1_B \succ_1 2_F2_B$. Both cases violate sd-weak-envy-freeness.

Then we consider the cases with the restriction that two of v, w, y, z are 0, and we list all the cases with possible combination of them as follows and briefly explain why they fail to meet the restriction.

$v \neq 0, y \neq 0$: there exists a generalized cycle $\{(2_F2_B, 2_F1_B), (2_F1_B, 2_F2_B)\}$.
$v \neq 0, w \neq 0$: agent 1 envies agent 2 due to the fact that $Q_1 \neq Q_2$ and $Q_1 \succeq_2^{sd} Q_2$.
$y \neq 0, z \neq 0$: agent 2 envies agent 1 due to the fact that $Q_2 \neq Q_1$ and $Q_2 \succeq_1^{sd} Q_1$.
$w \neq 0, z \neq 0$: there exists a generalized cycle $\{(1_F1_B, 1_F2_B), (1_F2_B, 1_F1_B)\}$.

By Proposition 2 (1), the existence of generalized cycle means violating sd-efficiency, which is also sd-weak-efficiency in the MTRA here because there are only two agents. Similar to Theorem 1, we can extend this case with contradiction to the one with $d \geq 2$ types and $n > 2$ agents. Therefore, we can conclude that such a mechanism f does not exist. □

Proof of Lemma 1

Proof First we show that if q satisfies the condition (i), it also satisfies the condition (ii) because from the Eq. (2) which computes the shares of bundles assigned by LexiPS, we have that for any $h < i$ and $\mathbf{w} \in D_1 \times \cdots \times D_h$,

$$\sum_{\mathbf{x} \in Z_{\mathbf{w}}} p_{\mathbf{x}} = \prod_{g \leq h} p^g_{w_g} = \prod_{g \leq h} q^g_{w_g} = \sum_{\mathbf{x} \in Z_{\mathbf{w}}} q_{\mathbf{x}}.$$

Then we prove the lemma when q satisfies the condition (ii). By the condition, we have that for any $h < i$,

$$\begin{aligned} \sum_{\mathbf{x} \in \{\mathbf{x} \in Z_{(y_g)_{g \leq h-1}} | x_h \succ^h y_h\}} p_{\mathbf{x}} &= \sum_{x_h \succ^h y_h} \sum_{\mathbf{x} \in Z_{(y_1,\ldots,y_{h-1},x_h)}} p_{\mathbf{x}} \\ &= \sum_{x_h \succ^h y_h} \sum_{\mathbf{x} \in Z_{(y_1,\ldots,y_{h-1},x_h)}} q_{\mathbf{x}} = \sum_{\mathbf{x} \in \{\mathbf{x} \in Z_{(y_g)_{g \leq h-1}} | x_h \succ^h y_h\}} q_{\mathbf{x}} \end{aligned} \tag{18}$$

Besides, since $\succ$ is a lexicographic preference, we can take apart the upper contour set $U(\succ, \mathbf{y})$ and have that:

$$\begin{aligned}\sum_{\mathbf{x}\in U(\succ,\mathbf{y})} p_{\mathbf{x}} = &\sum_{\mathbf{x}\in\{\mathbf{x}|x_1\succ^1 y_1\}} p_{\mathbf{x}} + \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_1)}|x_2\succ^2 y_2\}} p_{\mathbf{x}} \\ &+ \cdots + \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_h)_{h\le i-2}}|x_{i-1}\succ^{i-1} y_{i-1}\}} p_{\mathbf{x}} + \sum_{\mathbf{x}\in Z_{(y_h)_{h\le i-1}}\cap U(\succ,\mathbf{y})} p_{\mathbf{x}}\end{aligned} \tag{19}$$

We can derive a similar equation for q similar to the Eq. (19). Let $S = Z_{(y_h)_{h\le i-1}} \cap U(\succ, \mathbf{y})$. With Eq. (18), we see that if $\sum_{\mathbf{x}\in S} p_{\mathbf{x}} \ge \sum_{\mathbf{x}\in S} q_{\mathbf{x}}$, i.e. the inequality (3), then we have $\sum_{\mathbf{x}\in U(\succ,\mathbf{y})} p_{\mathbf{x}} \ge \sum_{\mathbf{x}\in U(\succ,\mathbf{y})} q_{\mathbf{x}}$. In the same way, if $\sum_{\mathbf{x}\in U(\succ,\mathbf{y})} p_{\mathbf{x}} \ge \sum_{\mathbf{x}\in U(\succ,\mathbf{y})} q_{\mathbf{x}}$, then we have $\sum_{\mathbf{x}\in S} p_{\mathbf{x}} \ge \sum_{\mathbf{x}\in S} q_{\mathbf{x}}$ by the Eqs. (18) and (19). Together we prove the lemma. □

Proof of Claim 1 in Theorem 2

Proof We prove the claim for each individual agent j. For ease of exposition, we label the types as $1 \rhd_j 2 \rhd_j \cdots \rhd_j d$. Then we need to prove that $Q^i_j = P^i_j$ if $Q^i \succeq^{sd}_j P^i$ for agent j. Suppose for the sake of contradiction that $Q^i_j \ne P^i_j$. Let N_i denote the set of agents who consume items of type i in Phase i and N'_i denote agents consume items of type i after Phase i. It is easy to see that $j \in N_i$. Given P^i_j, we use $P^i_{N_i} = (P^i_j)_{j\in N_i}$ to denote the partial assignment for agents in N_i. Now, we try to construct Q^i_j from P^i_j by transferring shares of bundles among agents. Notice that agent $k \in N \setminus (N_i \cup N'_i)$ who obtains items of type i before Phase i does not trade shares with agents in $N_i \cup N'_i$ because it means that $i \rhd_k i_k$ and we have that $Q^i_k = P^i_k$ by the condition. Therefore, in order to make $Q^i_j \ne P^i_j$, we just consider the share transferring among agents in $N_i \cup N'_i$ containing j. Then we show that any possible way of share transferring leads to a contradiction:

(i) If the share transferring only involves agents in N_i, then we have $Q^i_{N_i} \succeq^{sd} P^i_{N_i}$. However, from Algorithm 1, we learn that agents in N_i obey the rule of PS when consuming the items of type i left in Phase i, and therefore we have that $P^i_{N_i}$ is sd-efficient with the available items of type i in Phase i. With this and $Q^i_{N_i} \succeq^{sd} P^i_{N_i}$, we have that $Q^i_{N_i} = P^i_{N_i}$ where $j \in N_i$, a contradiction to $Q^i_j \ne P^i_j$.

(ii) Suppose that the share transferring also involves agents in N'_i. Let o'_i be the least preferred item agent j gets in P^i according to $\succ^i_j$. Then the best items of type i w.r.t. $\succ^i_j$ that agents in N'_i may have in P^i is o'_i because from Algorithm 1, we learn that agents in N'_i consume items of type i after agent j. By $Q^i \succeq^{sd}_j P^i$ and the assumption $Q^i_j \ne P^i_j$, there exists o_i such that $p^i_{j,o_i} < q^i_{j,o_i}$ and $\sum_{o\succ^i_j o_i} p^i_{j,o} = \sum_{o\succ^i_j o_i} q^i_{j,o}$. If $o_i \succ^i_j o'_i$, then the extra share of o_i in q^i_j comes from agents in N_i, which is a contradiction to sd-efficiency of $P^i_{N_i}$ as in the case (i). If $o_i = o'_i$ or $o'_i \succ^i_j o_i$, then we see that $\sum_{o\in U(\succ^i_j,o_i)} q^i_{j,o} > \sum_{o\in U(\succ^i_j,o_i)} p^i_{j,o} = \sum_{o\in D_i} p^i_{j,o} = 1$, which is a contradiction.

□

Proof of Claim 2 in Theorem 2

Proof Let $\mathbf{y} = (y_g)_{g\le d}$ be an arbitrary bundle. We define some notations with the condition that $p^h = q^h$ for $h \ne i$. Let $\alpha = \prod_{h<i} p^h_{y_i} = \prod_{h<i} q^h_{y_i}$. For $i < h < d$, we define

$$\beta_h = \prod_{i<g<h} p^g_{y_g} \cdot \sum_{o \succ^h y_h} p^h_o = \prod_{i<g<h} q^g_{y_g} \cdot \sum_{o \succ^h y_h} q^h_o.$$

Specially for type d,

$$\beta_d = \prod_{i<g<d} p^g_{y_g} \cdot \sum_{o \in U(\succ^d, y_d)} p^d_o = \prod_{i<g<d} q^g_{y_g} \cdot \sum_{o \in U(\succ^d, y_d)} q^d_o.$$

Since $\succ$ is lexicographic, we can deconstruct the shares over $\mathscr{D}' = \{\mathbf{x} \in Z_{(y_g)_{g\le i-1}} | \mathbf{x} \in U(\succ, \mathbf{y})\}$ as follows:

$$\begin{aligned}\sum_{\mathbf{x}\in\mathscr{D}'} p_{\mathbf{x}} = & \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le i-1}} | x_i \succ^i y_i\}} p_{\mathbf{x}} + \dots \\ & + \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le d-2}} | x_{d-1} \succ^{d-1} y_{d-1}\}} p_{\mathbf{x}} + \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le d-1}} | x_d \in U(\succ^d, y_d)\}} p_{\mathbf{x}}.\end{aligned} \tag{20}$$

From the Eq. (2) which computes the shares of bundles assigned by LexiPS, we have that for type i,

$$\sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le i-1}} | x_i \succ^i y_i\}} p_{\mathbf{x}} = \prod_{g<i} p^g_{y_i} \cdot \sum_{o \succ^i y_i} p^i_o = \alpha \cdot \sum_{o \succ^i y_i} p^i_o,$$

for $i < h < d$,

$$\sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le h-1}} | x_h \succ^h y_h\}} p_{\mathbf{x}} = \prod_{g<h} p^g_{y_g} \cdot \sum_{o \succ^h y_h} p^h_o = \alpha \cdot p^i_{y_i} \cdot \beta_h,$$

and for type d,

$$\sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le d-1}} | x_d \in U(\succ^d, y_d)\}} p_{\mathbf{x}} = \prod_{g<d} p^g_{y_g} \cdot \sum_{o \in U(\succ^d, y_d)} p^d_o = \alpha \cdot p^i_{y_i} \cdot \beta_d.$$

It is easy to see that we can derive similar equations for q to the ones above. Then, we can rewrite the Eq. (20) as $\sum_{\mathbf{x}\in\mathscr{D}'} p_{\mathbf{x}} = \alpha \cdot (\sum_{o\succ^i y_i} p^i_o + p^i_{y_i} \cdot \sum_{i<h\le d} \beta_h)$, and it follows that $\sum_{\mathbf{x}\in\mathscr{D}'} q_{\mathbf{x}} = \alpha \cdot (\sum_{o\succ^i y_i} q^i_o + q^i_{y_i} \cdot \sum_{i<h\le d} \beta_h)$ similarly for q. Because $p^i \succeq^{sd} q^i$, which means $\sum_{o\succ^i y_i} p^i_o \ge \sum_{o\succ^i y_i} q^i_o$ and $\sum_{o\in U(\succ^i, y_i)} p^i_o \ge \sum_{o\in U(\succ^i, y_i)} q^i_o$, we have that $\sum_{\mathbf{x}\in\mathscr{D}'} p_{\mathbf{x}} \ge \sum_{\mathbf{x}\in\mathscr{D}'} q_{\mathbf{x}}$, i.e.

$$\sum_{\mathbf{x}\in Z_{(y_g)_{g\le i-1}} \cap U(\succ, \mathbf{y})} p_{\mathbf{x}} \ge \sum_{\mathbf{x}\in Z_{(y_g)_{g\le i-1}} \cap U(\succ, \mathbf{y})} q_{\mathbf{x}}.$$

By Lemma 1, we have that $\sum_{\mathbf{x}\in U(\succ, \mathbf{y})} p_{\mathbf{x}} \ge \sum_{\mathbf{x}\in U(\succ, \mathbf{y})} q_{\mathbf{x}}$ for any $\mathbf{y}$, i.e. $p \succeq^{sd} q$. □

Proof of Proposition 2 (1)

Proof The proof involves showing that any fractional assignment which is not sd-efficient admits a generalized cycle. Let P be such a fractional assignment for a given MTRA. Then, there exists another fractional assignment $Q \neq P$ such that $Q \succeq^{sd} P$. We show that the set of tuples which shows the differences between P and Q is a generalized cycle on P.

Let $N' = \{j \in N | P_j \neq Q_j\} \subseteq N$, and it follows that $Q \succeq_j^{sd} P$ and $Q_j \neq P_j$ for any $j \in N'$. Let C be the set of tuples $\{(\mathbf{x}, \mathbf{y})|$ For some $j \in N', \mathbf{x} \succ_j \mathbf{y}, q_{j,\mathbf{x}} > p_{j,\mathbf{x}}, q_{j,\mathbf{y}} < p_{j,\mathbf{y}}\}$. At a high level, we can learn all of the differences in shares of bundles between the assignments P and Q from C. First we prove the following claim:

Claim 6 For every agent $j \in N'$, there exists a bundle $\mathbf{x}^j$ such that $q_{j,\mathbf{x}^j} > p_{j,\mathbf{x}^j}$ and for any $\mathbf{y}$ with $q_{j,\mathbf{y}} < p_{j,\mathbf{y}}$, $\mathbf{x}^j \succ_j \mathbf{y}$.

We prove it for an arbitrary agent j. Suppose for the sake of contradiction that there exists a bundle $\mathbf{y}$ such that $q_{j,\mathbf{y}} < p_{j,\mathbf{y}}$ and for any $\mathbf{z} \succ_j \mathbf{y}$, $q_{j,\mathbf{z}} \leq p_{j,\mathbf{z}}$. This implies that $\sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} q_{j,\mathbf{x}} < \sum_{\mathbf{x} \in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}}$, which is a contradiction to our assumption that $Q \succeq^{sd} P$.

Then we show that C is not empty. Consider an agent $j \in N'$. If $q_{j,\mathbf{x}} \geq p_{j,\mathbf{x}}$ and any $\mathbf{x} \in \mathscr{D}$, then $P_j = Q_j$, a contradiction to the fact that $j \in N'$. Thus there exists $\mathbf{y}$ with $q_{j,\mathbf{y}} < p_{j,\mathbf{y}}$, which means that there is a tuple $(\mathbf{x}^j, \mathbf{y}) \in C$ according to Claim 6. Therefore, $C \neq \emptyset$ since $P \neq Q$. For any $(\mathbf{x}, \mathbf{y}) \in C$, we have that $(\mathbf{x}, \mathbf{y}) \in Imp(P)$, due to the fact that $p_{j,\mathbf{y}} > q_{j,\mathbf{y}} \geq 0$ by our construction of C, which implies that $C \subseteq Imp(P)$.

Suppose for sake of contradiction that C is not a generalized cycle. Then we can find an item $o \in \mathbf{w}$ where $\mathbf{w}$ is the left component of some tuple in C such that o is never in the right component of any tuple in C. Then, we have that $q_{j,\mathbf{w}} \geq p_{j,\mathbf{w}}$ for any $j \in N'$. Otherwise, if for some agent $k \in N'$, $p_{k,\mathbf{w}} > q_{k,\mathbf{w}} \geq 0$, then we have that there exists $\mathbf{y}^k \succ_k \mathbf{w}$ by Claim 6, and therefore $(\mathbf{y}^k, \mathbf{w}) \in C$ due to $p_{k,\mathbf{w}} > 0$, which is a contradiction. Specifically, for any $(\mathbf{w}, \mathbf{z}) \in C$, there exists some $l \in N'$ such that $q_{l,\mathbf{w}} > p_{l,\mathbf{w}}$, because otherwise we have that $q_{j,\mathbf{w}} = p_{j,\mathbf{w}}$ for any $j \in N'$ and therefore the tuple $(\mathbf{w}, \mathbf{z})$ is not in C. We also have that $q_{j,\mathbf{w}} = p_{j,\mathbf{w}}$ trivially for $j \notin N'$ by the assumption. We note that the conclusions above about $\mathbf{w}$ also works for other bundles containing o which are left components of some tuples in C. For any bundle $\mathbf{y}$ with $o \in \mathbf{y}$ which does not occur in any tuple in C, we have that $q_{j,\mathbf{y}} = p_{j,\mathbf{y}}$ trivially for $j \in N$ by our construction of C.

Together, we have that $\sum_{j \in N, o \in \mathbf{x}} q_{j,\mathbf{x}} > \sum_{j \in N, o \in \mathbf{x}} p_{j,\mathbf{x}} = 1$, a contradiction to our assumption that Q is a fractional assignment. Thus C is a generalized cycle on P which is not sd-efficient. □

Proof of Claim 3 of Proposition 2

Proof Recall that for each agent $j \in N'$, the bundle $\mathbf{x}^j$ satisfies the condition that $q_{j,\mathbf{x}^j} > p_{j,\mathbf{x}^j}$ and $q_{j,\mathbf{x}} = p_{j,\mathbf{x}}$ for $\mathbf{x} \succ_j \mathbf{x}^j$. Suppose for the sake of contradiction that the claim is not true for some agent j and item $o \in \mathbf{x}^j$, which means that for any $k \in N'$ and $\mathbf{y}$ with $\mathbf{x}^k \succ_k \mathbf{y}$ and $o \in \mathbf{y}$, we have $p_{k,\mathbf{y}} = 0$. We note that j and k here may refer to the same agent. We split the set of bundles containing o into four parts, and we have that:

$$\sum_{o \in \mathbf{x}, k \in N} p_{k,\mathbf{x}} = \sum_{o \in \mathbf{x}, k \in N \setminus N'} p_{k,\mathbf{x}} + \sum_{o \in \mathbf{x}, k \in N', \mathbf{x} \succ_k \mathbf{x}^k} p_{k,\mathbf{x}} + \sum_{o \in \mathbf{x}^k, k \in N'} p_{k,\mathbf{x}^k} + \sum_{o \in \mathbf{x}, k \in N', \mathbf{x}^k \succ_k \mathbf{x}} p_{k,\mathbf{x}^k}$$

A similar equation can be derived for $\sum_{o \in \mathbf{x}, k \in N} q_{k,\mathbf{x}}$. The first part is the bundles containing o held by agents not in N', the shares of which are equal in P and Q. The remaining parts are the bundles containing o hold by N'. For each agent $k \in N'$, we split bundles containing o into three parts and compare their shares below:

(i) The bundles which are preferred over $\mathbf{x}^k$ by k, i.e. $\{\mathbf{x}|\mathbf{x} \succ_k \mathbf{x}^k\}$: Due to the fact that $\mathbf{x}$ satisfies $\mathbf{x} \succ_k \mathbf{x}^k$, we have $q_{k,\mathbf{x}} = p_{k,\mathbf{x}}$ by the selection of $\mathbf{x}^k$.

(ii) The bundle $\mathbf{x}^k$ that contains o, if it exists: We note that not all the bundles $\mathbf{x}^k$ with $k \in N'$ contains o. However, at least for agent j, we have that $\mathbf{x}^j$ contains o and $q_{j,\mathbf{x}^j} > p_{j,\mathbf{x}^j}$.

(iii) The bundles to which $\mathbf{x}^k$ is preferred by k, i.e. $\{\mathbf{x}|\mathbf{x}^k \succ_k \mathbf{x}\}$. Due to the fact that $\mathbf{x}$ satisfies $\mathbf{x}^k \succ_k \mathbf{x}$, we have that $q_{k,\mathbf{x}} \geq p_{k,\mathbf{x}} = 0$ by our assumption.

Therefore, we have that $1 = \sum_{o\in\mathbf{x},k\in N} p_{k,\mathbf{x}} < \sum_{o\in\mathbf{x},k\in N} q_{k,\mathbf{x}} = 1$, which is a contradiction. It means that for any agent j and item $o \in \mathbf{x}^j$, there exist $k \in N'$ and $\mathbf{y}$ such that $\mathbf{x}^k \succ_k \mathbf{y}$, $o \in \mathbf{y}$ and $p_{k,\mathbf{y}} > 0$. □

Proof of Proposition 3

Proof Consider an MTRA (N, M) and any preference profile R. Throughout, we use P to refer to MPS(R). We first give a few observations about MPS which are helpful for understanding the following proof. We know that MPS executes multiple rounds which come to an end when some items are exhausted and we label all the rounds by the time at which they end. Let j be an arbitrary agent. Here we consider the set of rounds $\{r_{k_1}, r_{k_2}, \dots\}$ such that at the end of each round r_{k_b} in it, agent j stops consuming a bundle. We note that these rounds are not necessarily continuous because agent j may not change her current most preferred bundles at the end of some rounds. W.l.o.g. let $r_{k_b} < r_{k_{b'}}$ if $b < b'$.

Let $\mathbf{x}^{k_b}$ denote the bundle consumed by j at round r_{k_b} of MPS. Let $t_{k_0} = 0$, and for any round r_{k_b}, let t_{k_b} be the units of time elapsed from the start of the mechanism till the end of round r_{k_b}. Then, by construction of MPS, we have that $p_{j,\mathbf{x}^{k_b}} = t_{k_b} - t_{k_{b-1}}$ for any round r_{k_b} with $b > 1$. Specially, when $b = 1$, $p_{j,\mathbf{x}^{k_1}} = t_{k_1} - t_{k_0}$ trivially. This implies that

$$t_{k_b} = t_{k_b} - t_{k_0} = \sum_{b'=1}^{b} (t_{k_{b'}} - t_{k_{b'-1}}) = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{x}^{k_b})} p_{j,\mathbf{x}}. \tag{21}$$

For any round r_{k_b} and $\mathbf{y}$ such that $\mathbf{x}^{k_b} \succ_j \mathbf{y} \succ_j \mathbf{x}^{k_{b+1}}$, $\mathbf{y}$ is not consumed by j, i.e. $p_{j,\mathbf{y}} = 0$. Therefore, it must hold that $\mathbf{y}$ is unavailable by the end of round r_{k_b}. Let t denote the time at which $\mathbf{y}$ becomes unavailable. Then,

$$t \leq t_{k_b} = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{x}^{k_b})} p_{j,\mathbf{x}} = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}}. \tag{22}$$

With these observations we begin the proof. Suppose for the sake of contradiction that there is a pair of agents j and k such $P_j \not\succeq_j^{sd} P_k$. Then, there exists a bundle $\mathbf{y}$ which satisfies that $\sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{k,\mathbf{x}} > \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}}$.

Let $t = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}}$ and $t' = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{k,\mathbf{x}}$. It is easy to see that $t < t'$ by the assumption. The rest of the proof involves showing that due to the construction of MPS, $t' \leq t$, contradicting our assumption.

Now, let $\mathbf{z}$ be the least preferred bundle in the set $\{\mathbf{x} \in U(\succ_j, \mathbf{y})|p_{k,\mathbf{x}} > 0\}$ for agent k. Such a bundle $\mathbf{z}$ must exist. Otherwise, $p_{k,\mathbf{x}} = 0$ for any $\mathbf{x} \in U(\succ_j, \mathbf{y})$, which implies that $t' = 0 \leq t$, a contradiction.

Let $t_{\mathbf{z}}$ be the time at which $\mathbf{z}$ becomes unavailable. Due to $U(\succ_j, \mathbf{z}) \subseteq U(\succ_j, \mathbf{y})$, we can deduce that $t_{\mathbf{z}} \leq \sum_{\mathbf{x}\in U(\succ_j,\mathbf{z})} p_{j,\mathbf{x}} \leq \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}} = t$ by the inequality (22). Also, we have that $t_{\mathbf{z}} = \sum_{\mathbf{x} \in U(\succ_k, \mathbf{z})} p_{k,\mathbf{x}}$ by the Eq. (21). By the selection of $\mathbf{z}$, we have that $\{\mathbf{x} \in U(\succ_j, \mathbf{y}) | p_{k,\mathbf{x}} > 0\} \subseteq U(\succ_k, \mathbf{z})$. Therefore, we can deduce that

$$t' = \sum_{\mathbf{x} \in \{\mathbf{x} \in U(\succ_j, \mathbf{y}) | p_{k,\mathbf{x}} > 0\}} p_{k,\mathbf{x}} \leq \sum_{\mathbf{x} \in U(\succ_k, \mathbf{z})} p_{k,\mathbf{x}} = t_{\mathbf{z}}.$$

This implies $t' \leq t_{\mathbf{z}} \leq t$, which is a contradiction to the assumption. Therefore we have that $P_j \succeq_j^{sd} P_k$ for any agents j and k, which completes the proof. □

Proof of Theorem 5

Proof Since in Proposition 2 we show that lexi-efficiency is equivalent to no-generalized-cycle condition, we can prove the theorem by showing that an assignment satisfies no-generalized-cycle if and only if the assignment is the output of an eating algorithm.

Sufficiency This proof is similar to the idea of proof for Theorem 5 in [42]. Let P be the output of an eating algorithm given a preference profile R. Suppose for the sake of contradiction that P admits a generalized cycle C. We use $t(o)$ to stand for the time when o is exhausted in the eating algorithm and use *Seq* to denote a partial order on M such that o *Seq* o' if $t(o) \leq t(o')$ for any pair of items o and o'.

Let $M' = \{o \in M | o \in \mathbf{x}, (\mathbf{x}, \mathbf{y}) \in C\}$ and $o' \in M'$ be the item satisfying o' *Seq* o for any $o \in M'$. By the definition of generalized cycles, there is an improvable tuple $(\mathbf{x}, \mathbf{y}) \in C$ such that $o' \in \mathbf{y}$. It means that there exists an agent $j \in N$ such that $\mathbf{x} \succ_j \mathbf{y}$ and $p_{j,\mathbf{y}} > 0$. Hence, when agent j starts to consume $\mathbf{y}$, the bundle $\mathbf{x}$ is unavailable with an item $o \in \mathbf{x}$ which is exhausted. We note that $o \in M'$ due to the fact that $(\mathbf{x}, \mathbf{y}) \in C$. Then, we have that $t(o) < t(o')$ and therefore o *Seq* o', a contradiction to the selection of o'.

Necessity Let R be an arbitrary preference profile and P be any assignment satisfying the no-generalized-cycle condition w.r.t. R. For convenience, we define some quantities to represent the state during the execution of a member of the family of eating algorithms at each round. For ease of exposition, we use s to denote a round and $s = 0$ represents the initial state before the start of execution. Let $M^0 = M$ and $\mathscr{D}^0 = \mathscr{D}$. We define recursively that $B^s = \{o \in M^{s-1} | \text{there are no } \mathbf{x}, \mathbf{y} \in \mathscr{D}^{s-1} \text{ with } o \in \mathbf{y} \text{ and } (\mathbf{x}, \mathbf{y}) \in Imp(P)\}$, $M^s = M^{s-1} \setminus B^s$, and $\mathscr{D}^s = \{\mathbf{x} \in \mathscr{D} | \text{for every } o \in \mathbf{x}, o \in M^s\}$ be the available bundles in M^s. We note that $B^s \neq \emptyset$ for any s with $M^{s-1} \neq \emptyset$. Otherwise, for any $o \in M^{s-1}$, there exists $(\mathbf{x}, \mathbf{y}) \in Imp(P)$ with $\mathbf{x}, \mathbf{y} \in \mathscr{D}^{s-1}$ and $o \in \mathbf{y}$. We note that for any $\mathbf{x} \in \mathscr{D}^{s-1}$ and $o \in \mathbf{x}$, it follows that $o \in M^{s-1}$, and therefore $Imp(P)$ is a generalized cycle, a contradiction. We use $u = \min\{s | M^s = \emptyset\}$ to denote the round where every item is exhausted and $N(\mathbf{x}, \mathscr{D}^s)$ to refer to the set of agents who prefer $\mathbf{x}$ best in the available bundles $\mathscr{D}^s$. We have the following claim with these new notations.

Claim 7 For an assignment P satisfying the no-generalized-cycle condition and any $o \in B^s$,

$$\sum_{o \in \mathbf{x}, \mathbf{x} \in \mathscr{D}^{s-1}} \sum_{j \in N(\mathbf{x}, \mathscr{D}^{s-1})} p_{j,\mathbf{x}} + \sum_{o \in \mathbf{x}, \mathbf{x} \notin \mathscr{D}^{s-1}} \sum_{j \in N} p_{j,\mathbf{x}} = \sum_{o \in \mathbf{x}, \mathbf{x} \in \mathscr{D}} \sum_{j \in N} p_{j,\mathbf{x}} = 1.$$

We can prove the claim with the fact that for any $o \in B^s$,

$$\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^{s-1}}\sum_{j\in N(\mathbf{x},\mathscr{D}^{s-1})}p_{j,\mathbf{x}}+\sum_{o\in\mathbf{x},\mathbf{x}\notin\mathscr{D}^{s-1}}\sum_{j\in N}p_{j,\mathbf{x}}\le\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}}\sum_{j\in N}p_{j,\mathbf{x}}=1,$$

and the equality must hold. Otherwise, there exist $\mathbf{x},\mathbf{y}\in\mathscr{D}^{s-1}$ and $k\in N(\mathbf{x},\mathscr{D}^{s-1})$ such that $\mathbf{x}\succ_k\mathbf{y}$, $o\in\mathbf{y}$ and $p_{k,\mathbf{y}}>0$, which implies that $o\notin B^s$, a contradiction.

With the observation in Claim 7, we specify an instance of Algorithm 3 with the following eating speed functions ω_j: for any $s\le u$ and $\frac{s-1}{u}\le t\le\frac{s}{u}$,

$$\omega_j(t)\overset{\text{def}}{=}\begin{cases}u\cdot p_{j,\mathbf{x}}, & o\in\mathbf{x}, o\in B^s\text{ and } j\in N(\mathbf{x},\mathscr{D}^{s-1}),\\ 0, & \text{otherwise.}\end{cases}$$

From the design of algorithm, we know that items in B^s decide which bundles in $\mathscr{D}^{s-1}$ are consumed in the round s, and these items are not consumed after the round s. Note that $j\in N(\mathbf{x},\mathscr{D}^{s-1})$ implies $\mathbf{x}\in\mathscr{D}^{s-1}$.

We claim that the algorithm specified by the eating speed functions $\omega=(\omega_j)_{j\le n}$ above outputs P for the MTRA with preference profile R. Let Q be the output of the eating algorithm. We prove that $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$ for any agent j and the bundle $\mathbf{x}\in\mathscr{D}^{s-1}$ containing an item $o\in B^s$ by induction on the round s.

Base case We prove that $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$ for any $j\in N$, $o\in B^1$ and bundle $\mathbf{x}$ with $o\in\mathbf{x}$. For $j\in N(\mathbf{x},\mathscr{D}^0)$ where $\mathbf{x}$ contains an item $o\in B^1$, we know that agent j consumes bundle $\mathbf{x}$ which is available during the period $[0,\frac{1}{u}]$, and therefore $q_{j,\mathbf{x}}\ge\frac{1}{u}\cdot u\cdot p_{j,\mathbf{x}}=p_{j,\mathbf{x}}$. By Claim 7, we have that

$$1=\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^0}\sum_{j\in N(\mathbf{x},\mathscr{D}^0)}p_{j,\mathbf{x}}\le\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^0}\sum_{j\in N(\mathbf{x},\mathscr{D}^0)}q_{j,\mathbf{x}}\le 1,$$

which means that $\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^0}\sum_{j\in N(\mathbf{x},\mathscr{D}^0)}p_{j,\mathbf{x}}=\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^0}\sum_{j\in N(\mathbf{x},\mathscr{D}^0)}q_{j,\mathbf{x}}$. With $q_{j,\mathbf{x}}\ge p_{j,\mathbf{x}}$ for any $j\in N(\mathbf{x},\mathscr{D}^0)$ where $\mathbf{x}$ contains $o\in B^1$, the equation implies that $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$, and we also have that $q_{k,\mathbf{x}}=0$ for $k\notin N(\mathbf{x},\mathscr{D}^0)$ by Claim 7. Together they means that $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$ for any agent $j\in N$, $o\in B^1$ and bundle $\mathbf{x}$ with $o\in\mathbf{x}$.

Inductive step Assume $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$ for any $j\in N$, $o\in\bigcup_{r\le s}B^r$ and $\mathbf{x}\in\mathscr{D}$ with $o\in\mathbf{x}$. We prove that $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$ for any $j\in N$, $o\in B^{s+1}$ and $\mathbf{x}\in\mathscr{D}^s$ with $o\in\mathbf{x}$. If $\mathbf{x}\notin\mathscr{D}^s$, then there is an item $o'\in\mathbf{x}$ satisfying $o'\in\bigcup_{r\le s}B^r$, and therefore we have that $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$ by the assumption. Then we show that $p_{j,\mathbf{x}}=q_{j,\mathbf{x}}$ for any $j\in N(\mathbf{x},\mathscr{D}^s)$, $o\in B^{s+1}$ and $\mathbf{x}\in\mathscr{D}^s$ with $o\in\mathbf{x}$. By the assumption, any $o'\in M^s$ is available with the supply of at least $\sum_{o'\in\mathbf{y},\mathbf{y}\in\mathscr{D}^s}\sum_{j\in N(\mathbf{y},\mathscr{D}^s)}p_{j,\mathbf{x}}$ at time $t=\frac{s}{u}$. From the algorithm, we know that agent j consumes $\mathbf{x}$ during $[\frac{s}{u},\frac{s+1}{u}]$, and therefore $q_{j,\mathbf{x}}\ge\frac{1}{u}\cdot u\cdot p_{j,\mathbf{x}}=p_{j,\mathbf{x}}$. Hence,

$$\begin{aligned}\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^s}\sum_{j\in N(\mathbf{x},\mathscr{D}^s)}p_{j,\mathbf{x}}+\sum_{o\in\mathbf{x},\mathbf{x}\notin\mathscr{D}^s}\sum_{j\in N}p_{j,\mathbf{x}}&=\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^s}\sum_{j\in N(\mathbf{x},\mathscr{D}^s)}p_{j,\mathbf{x}}+\sum_{o\in\mathbf{x},\mathbf{x}\notin\mathscr{D}^s}\sum_{j\in N}q_{j,\mathbf{x}}\\&\le\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^s}\sum_{j\in N(\mathbf{x},\mathscr{D}^s)}q_{j,\mathbf{x}}+\sum_{o\in\mathbf{x},\mathbf{x}\notin\mathscr{D}^s}\sum_{j\in N}q_{j,\mathbf{x}}\le 1.\end{aligned}$$

Then, by Claim 7, we have that

$$\sum_{o\in\mathbf{x},\mathbf{x}\in\mathscr{D}^s}\sum_{j\in N(\mathbf{x},\mathscr{D}^s)}q_{j,\mathbf{x}}+\sum_{o\in\mathbf{x},\mathbf{x}\notin\mathscr{D}^s}\sum_{j\in N}q_{j,\mathbf{x}}=1.$$

With $q_{j,\mathbf{x}} \geq p_{j,\mathbf{x}}$ for any $j \in N(\mathbf{x}, \mathscr{D}^s)$ where $\mathbf{x}$ contains $o \in B^{s+1}$, the equation implies that $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$, and we also have that $q_{k,\mathbf{x}} = 0$ for $k \notin N(\mathbf{x}, \mathscr{D}^s)$ by Claim 7. Together they means that $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$ for any $j \in N$, $o \in B^{s+1}$ and $\mathbf{x} \in \mathscr{D}^s$ with $o \in \mathbf{x}$.

By induction, we have that $p_{j,\mathbf{x}} = q_{j,\mathbf{x}}$ for any $j \in N$, $o \in \bigcup_{r \leq u} B^r$ and $\mathbf{x} \in \mathscr{D}$ with $o \in \mathbf{x}$. This also means that $P = Q$, and it follows that the eating algorithm specified by the eating speed functions $\omega = (\omega_j)_{j \leq n}$ exactly outputs the assignment P for the given MTRA with R. □

Proof of Claim 4 in Theorem 6

Proof We know that each agent performs the following step repeatedly in PS: consuming her most preferred and unexhausted item till it is exhausted. To prove the claim, we show that agents in MPS performs the same step in each type i as they do in PS applied to just the type i, i.e. for any $i \leq d, j \leq n$, agent j consumes her most preferred and unexhausted item in type i while consuming her most preferred and available bundle. In the following discussion, we use $(o', \mathbf{x} \backslash o)$ to denote the bundle replacing o with o' in $\mathbf{x}$.

W.l.o.g. we consider what agent j does in MPS for type i. At the beginning of MPS, the bundle consumed by agent j, denoted by $\mathbf{x}^1$, is her most preferred bundle w.r.t. $\succ_j$. By Definition 7, the most preferred bundle contains agent j's most preferred item in type i w.r.t. $\succ_j^i$. This means that when agent j consumes $\mathbf{x}^1$, she also consumes her most preferred item $o_1 = D_i(\mathbf{x}^1)$. When the consumption of $\mathbf{x}^1$ pauses, agent j turns to the bundle consumed after $\mathbf{x}^1$, denoted by $\mathbf{x}^2$. We note that $\mathbf{x}^2$ does not need to be the second preferred bundle w.r.t. $\succ_j$. There are two kinds of cases before consuming $\mathbf{x}^2$: (i) $D_h(\mathbf{x}^1)$ is exhausted, $h \neq i$, (ii) o_1 is exhausted.

We claim that $D_i(\mathbf{x}^2)$ is the most preferred and unexhausted item in type i just after $\mathbf{x}^1$ is unavailable for both cases. For the case (i), the agent j's most preferred item in type i is still o_1. We claim that $o_1 \in \mathbf{x}^2$. Otherwise, suppose that $o_2 = D_i(\mathbf{x}^2) \neq o_1$. Then $o_1 \succ_j^i o_2$. Because $D_h((o_1, \mathbf{x}^2 \backslash o_2)) = D_h(\mathbf{x}^2)$ for $h \rhd_j i$ and $o_1 = D_i((o_1, \mathbf{x}^1 \backslash o_2)) \succ_j^i D_i(\mathbf{x}^2) = o_2$, it follows that $(o_1, \mathbf{x}^2 \backslash o_2) \succ_j \mathbf{x}^2$, which is a contradiction to the fact that $\mathbf{x}^2$ is the most preferred and available bundle for agent j after $\mathbf{x}^1$ is unavailable. For the case (ii), let M_1 be the set of unexhausted items just after $\mathbf{x}^1$ is unavailable in MPS, and $o_2 \in M_1 \bigcap D_i$ be agent j's most preferred and unexhausted item in D_i w.r.t. $\succ_j^i$. We also note that o_2 does not need to be the second preferred item w.r.t. $\succ_j^i$. By Algorithm 2, we know that $o_2 = D_i(\mathbf{x}^2)$, and we can obtain that $\mathbf{x}^2 \succ_j (o_2', \mathbf{x}^2 \backslash o_2)$ if $o_2' \in M_1 \bigcap D_i$ and $o_2' \neq o_2$.

The claim in the previous paragraph can be applied to the general case when $\mathbf{x}^k$ is unavailable and turns to $\mathbf{x}^{k+1}$, and we have that $D_i(\mathbf{x}^{k+1})$ is always the most preferred and unexhausted item in type i just after $\mathbf{x}^k$ is unavailable. We note again that $\mathbf{x}^k$ does not need to be the k-th preferred item w.r.t. $\succ_j$, and $\mathbf{x}^{k+1}$ only refers to the bundle consumed after $\mathbf{x}^k$. From the argument above, we can observe that agent j consumes the bundle containing the most preferred and unexhausted item in type i. This is exactly what agent j does in PS of type i. This argument can be extended to any agent and any type. Thus, each single type fractional assignment w.r.t. P which is the output of MPS is the same as the one produced by PS of that type. □

Proof of Claim 5 in Theorem 6

Proof When $h \le i = 1$ i.e. $P_j^1 = Q_j^1$, the claim means that for $S_1 = \{\mathbf{x} | x_1 \in U(\succ_j^1, y_1)\}$,

$$\sum_{\mathbf{x}\in S_1} p_{j,\mathbf{x}} = \sum_{o\in U(\succ_j^1, y_1)} p_{j,o}^1 = \sum_{o\in U(\succ_j^1, y_1)} q_{j,o}^1 = \sum_{\mathbf{x}\in S_1} q_{j,\mathbf{x}},$$

which is trivially true.

Suppose that the claim is true for $h < i$ with $i > 1$. Given the conditions that $P^h = Q^h$ for $h \le i$ and $Q \succeq_j^{sd} P$, we have the following by the assumption: for any $h < i$,

$$\sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le h-1}} | x_h \succ_j^h y_h\}} p_{j,\mathbf{x}} = \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_g)_{g\le h-1}} | x_h \succ_j^h y_h\}} q_{j,\mathbf{x}}. \tag{23}$$

Then we prove the claim for i. Let $\mathbf{y}$ denote the least preferred bundle containing $y_1, \dots, y_i$. We can take apart $\sum_{\mathbf{x}\in U(\succ_j, \mathbf{y})} p_{j,\mathbf{x}}$, i.e. agent j's shares over $U(\succ_j, \mathbf{y})$ in P as:

$$\sum_{\mathbf{x}\in\{\mathbf{x}|x_1\succ_j^1 y_1\}} p_{j,\mathbf{x}} + \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_1)} | x_2 \succ_j^2 y_2\}} p_{j,\mathbf{x}} + \dots + \sum_{\mathbf{x}\in\{\mathbf{x}\in Z_{(y_h)_{h\le i-2}} | x_{i-1} \succ_j^{i-1} y_{i-1}\}} p_{j,\mathbf{x}} + \sum_{\mathbf{x}\in S_i} p_{j,\mathbf{x}}. \tag{24}$$

We have a similar equation for $\sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} q_{j,\mathbf{x}}$. We see that $\mathbf{w} \succ_j \mathbf{z}$ for any $\mathbf{w} \in U(\succ_j, \mathbf{y}) \setminus S_i$ and $\mathbf{z} \in S_i$. W.l.o.g. let $\mathbf{w}$ denote the least preferred bundle in $U(\succ_j, \mathbf{y}) \setminus S_i$. By the Eqs. (23) and (24), we have that $\sum_{\mathbf{x}\in U(\succ_j,\mathbf{w})} p_{j,\mathbf{x}} = \sum_{\mathbf{x}\in U(\succ_j,\mathbf{w})} q_{j,\mathbf{x}}$. With this and $\sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} p_{j,\mathbf{x}} \le \sum_{\mathbf{x}\in U(\succ_j,\mathbf{y})} q_{j,\mathbf{x}}$ implied by $Q \succeq_j^{sd} P$, we have that $\sum_{\mathbf{x}\in S_i} p_{j,\mathbf{x}} \le \sum_{\mathbf{x}\in S_i} q_{j,\mathbf{x}}$. By summing up each side over all the possible choices of $y_1, \dots, y_{i-1}$, we have that

$$\sum_{y_1}\sum_{y_2}\cdots\sum_{y_{i-1}}\sum_{\mathbf{x}\in S_i} p_{j,\mathbf{x}} = \sum_{x_i\in U(\succ_j^i, y_i)} p_{j,x_i}^i \le \sum_{x_i\in U(\succ_j^i, y_i)} q_{j,x_i}^i = \sum_{y_1}\sum_{y_2}\cdots\sum_{y_{i-1}}\sum_{\mathbf{x}\in S_i} q_{j,\mathbf{x}}.$$

With the condition that $P_j^i = Q_j^i$, we have that $\sum_{x_i\in U(\succ_j^i, y_i)} p_{j,x_i}^i = \sum_{x_i\in U(\succ_j^i, y_i)} q_{j,x_i}^i$. With this and $\sum_{\mathbf{x}\in S_i} p_{j,\mathbf{x}} \le \sum_{\mathbf{x}\in S_i} q_{j,\mathbf{x}}$, it follows that $\sum_{\mathbf{x}\in S_i} p_{j,\mathbf{x}} = \sum_{\mathbf{x}\in S_i} q_{j,\mathbf{x}}$ for S_i with any $y_1, \dots, y_i$. □

Acknowledgements The authors would like to thank the editors and anonymous reviewers for their helpful comments. LX acknowledges NSF #1453542 and #1716333 for support. YC acknowledges NSFC under Grants 61772035 and 61932001, and the National Key R&D Program under Grant 2020YFB1005702 for support. HW acknowledges NSFC under Grant 61972005 and the National Key R&D Program under Grants 2018YFB1003904 and 2018YFC1314200 for support.

References

1. Abdulkadiroğlu, A., & Sönmez, T. (1998). Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica, 66*(3), 689–702.
2. Abdulkadiroğlu, A., & Sönmez, T. (1999). House allocation with existing tenants. *Journal of Economic Theory, 88*, 233–260.
3. Athanassoglou, S., & Sethuraman, J. (2011). House allocation with fractional endowments. *International Journal of Game Theory, 40*(3), 481–513.

4. Aziz, H., & Kasajima, Y. (2017). Impossibilities for probabilistic assignment. *Social Choice and Welfare, 49*(2), 255–275.
5. Aziz, H., & Stursberg, P. (2014). A generalization of probabilistic serial to randomized social choice. In *Proceedings of the twenty-eighth AAAI conference on artificial intelligence* (pp. 559–565).
6. Birkhoff, G. (1946). Three observations on linear algebra. *Universidad Nacional de Tucuman, Revista, Serie A, 5,* 147–151.
7. Bogomolnaia, A. (2015). Random assignment: Redefining the serial rule. *Journal of Economic Theory, 158,* 308–318.
8. Bogomolnaia, A., & Heo, E. J. (2012). Probabilistic assignment of objects: Characterizing the serial rule. *Journal of Economic Theory, 147*(5), 2072–2082.
9. Bogomolnaia, A., & Moulin, H. (2001). A new solution to the random assignment problem. *Journal of Economic Theory, 100*(2), 295–328.
10. Bogomolnaia, A., Moulin, H., & Stong, R. (2005). Collective choice under dichotomous preferences. *Journal of Economic Theory, 122*(2), 165–184.
11. Booth, R., Chevaleyre, Y., Lang, J., Mengin, J., & Sombattheera, C. (2010). Learning conditionally lexicographic preference relations. In *Proceedings of the 2010 conference on ECAI 2010: 19th European conference on artificial intelligence* (pp 269–274).
12. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., & Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus statements. *Journal of Artificial Intelligence Research, 21,* 135–191.
13. Budish, E., Che, Y. K., Kojima, F., & Milgrom, P. (2013). Designing random allocation mechanisms: Theory and applications. *American Economic Review, 103*(2), 585–623.
14. Chatterji, S., & Liu, P. (2020). Random assignments of bundles. *Journal of Mathematical Economics, 87,* 15–30.
15. Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaitre, M., Maudet, N., et al. (2006). Issues in multiagent resource allocation. *Informatica, 30,* 3–31.
16. Cho, W. J. (2014). Impossibility results for parametrized notions of efficiency and strategy-proofness in exchange economies. *Games and Economic Behavior*, *86*, 26–39.
17. Cho, W. J. (2018). Probabilistic assignment: an extension approach. *Social Choice and Welfare*, *51*(1), 137–162.
18. Cloutier, J., Nyman, K. L., & Su, F. E. (2010). Two-player envy-free multi-cake division. *Mathematical Social Sciences, 59*(1), 26–37.
19. Fujita, E., Lesca, J., Sonoda, A., Todo, T., & Yokoo, M. (2018). A complexity approach for core-selecting exchange under conditionally lexicographic preferences. *Journal of Artificial Intelligence Research, 63,* 515–555.
20. Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., & Stoica, I. (2011). Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX conference on networked systems design and implementation, Boston, MA, USA* (pp. 323–336).
21. Ghodsi, A., Sekar, V., Zaharia, M., & Stoica, I. (2012). Multi-resource fair queueing for packet processing. In *Proceedings of the ACM SIGCOMM 2012 conference on applications, technologies, architectures, and protocols for computer communication* (pp. 1–12).
22. Grandl, R., Ananthanarayanan, G., Kandula, S., Rao, S., & Akella, A. (2015). Multi-resource packing for cluster schedulers. *ACM SIGCOMM Computer Communication Review, 44*(4), 455–466.
23. Hashimoto, T., & Hirata, D. (2011). Characterizations of the probabilistic serial mechanism. Available at SSRN https://ssrn.com/abstract=1601178. Retrieved on 18 Jan 2020.
24. Hashimoto, T., Hirata, D., Kesten, O., Kurino, M., & Ünver, M. U. (2014). Two axiomatic approaches to the probabilistic serial mechanism. *Theoretical Economics, 9*(1), 253–277.
25. Heo, E. J. (2014). Probabilistic assignment problem with multi-unit demands: A generalization of the serial rule and its characterization. *Journal of Mathematical Economics, 54,* 40–47.
26. Heo, E. J., & Yılmaz, Ö. (2015). A characterization of the extended serial correspondence. *Journal of Mathematical Economics, 59,* 102–110.
27. Hosseini, H., Igarashi, A., & Searns, A. (2020). Fair division of time: Multi-layered cake cutting. arXiv preprint arXiv:200413397.
28. Katta, A. K., & Sethuraman, J. (2006). A solution to the random assignment problem on the full preference domain. *Journal of Economic Theory, 131*(1), 231–250.
29. Kojima, F. (2009). Random assignment of multiple indivisible objects. *Mathematical Social Sciences, 57*(1), 134–142.
30. Lang, J., Mengin, J., & Xia, L. (2012). Aggregating conditionally lexicographic preferences on multi-issue domains. In *Proceedings of the 18th international conference on principles and practice of constraint programming (CP)* (pp. 973–987).

31. Lebert, N., Meunier, F., & Carbonneaux, Q. (2013). Envy-free two-player m-cake and three-player two-cake divisions. *Operations Research Letters, 41*(6), 607–610.
32. Mackin, E., & Xia, L. (2016). Allocating indivisible items in categorized domains. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence* (pp. 359–365).
33. Moulin, H. (2014). *Cooperative microeconomics: A game-theoretic introduction*. Princeton University Press.
34. Moulin, H. (2019). Fair division in the age of internet. *Annual Review of Economics*, *11*(1), 407–441.
35. Nguyen, T., Peivandi, A., & Vohra, R. (2016). Assignment problems with complementarities. *Journal of Economic Theory, 165,* 209–241.
36. Nyman, K., Su, F. E., & Zerbib, S. (2020). Fair division with multiple pieces. *Discrete Applied Mathematics, 283,* 115–122.
37. Saban, D., & Sethuraman, J. (2014). A note on object allocation under lexicographic preferences. *Journal of Mathematical Economics, 50,* 283–289.
38. Segal-Halevi, E. (2016). Fair division of land. Ph.D. thesis, Bar Ilan University, Computer Science Department.
39. Sikdar, S., Adalı, S., & Xia, L. (2017). Mechanism design for multi-type housing markets. In *Proceedings of the thirty-first AAAI conference on artificial intelligence* (pp. 684–690).
40. Sikdar, S., Adalı, S., & Xia, L. (2019). Mechanism design for multi-type housing markets with acceptable bundles. In *Proceedings of the thirty-third AAAI conference on artificial intelligence* (pp. 2165–2172).
41. Von Neumann, J. (1953). A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games, 2,* 5–12.
42. Wang, H., Sikdar, S., Guo, X., Xia, L., Cao, Y., & Wang, H. (2020). Multi-type resource allocation with partial preferences. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, pp. 2260–2267).
43. Xia, L., & Conitzer, V. (2010). Strategy-proof voting rules over multi-issue domains with restricted preferences. In Saberi, A. (ed) *Internet and Network Economics.* WINE 2010. Lecture Notes in Computer Science, Vol 6484. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-17572-5_33
44. Yilmaz, Ö. (2009). Random assignment under weak preferences. *Games and Economic Behavior, 66*(1), 546–558.
45. Yılmaz, Ö. (2010). The probabilistic serial mechanism with private endowments. *Games and Economic Behavior, 69*(2), 475–491.
46. Zhou, L. (1990). On a conjecture by gale about one-sided matching problems. *Journal of Economic Theory, 52*(1), 123–135.

Authors and Affiliations

Xiaoxi Guo[1] · Sujoy Sikdar[2] · Haibin Wang[1] · Lirong Xia[3] · Yongzhi Cao[1] · Hanpin Wang[1,4]

Xiaoxi Guo
guoxiaoxi@pku.edu.cn

Sujoy Sikdar
ssikdar@binghamton.edu

Haibin Wang
beach@pku.edu.cn

Lirong Xia
xialirong@gmail.com

Hanpin Wang
whpxhy@pku.edu.cn

[1] Key Laboratory of High Confidence Software Technologies (MOE), Department of Computer Science and Technology, Peking University, Beijing, China

[2] Department of Computer Science, Binghamton University, Binghamton, USA

[3] Department of Computer Science, Rensselaer Polytechnic Institute, Troy, USA

[4] School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China