On Fair Division under Heterogeneous Matroid Constraints *

Amitay Dror,[1] **Michal Feldman,**[2] **Erel Segal-Halevi** [3]
[1]Tel Aviv University
[2]Tel Aviv University, Microsoft Research
[3]Ariel University
amitaydr@gmail.com, mfeldman@tau.ac.il, erelsgl@gmail.com

Abstract

We study fair allocation of indivisible goods among additive agents with feasibility constraints. In these settings, every agent is restricted to get a bundle among a specified set of feasible bundles. Such scenarios have been of great interest to the AI community due to their applicability to real-world problems. Following some impossibility results, we restrict attention to matroid feasibility constraints that capture natural scenarios, such as the allocation of shifts to medical doctors, and the allocation of conference papers to referees.

We focus on the common fairness notion of envy-freeness up to one good (EF1). Previous algorithms for finding EF1 allocations are either restricted to agents with identical feasibility constraints, or allow free disposal of items. An open problem is the existence of EF1 complete allocations among heterogeneous agents, where the heterogeneity is both in the agents' feasibility constraints and in their valuations. In this work, we make progress on this problem by providing positive and negative results for different matroid and valuation types. Among other results, we devise poly-time algorithms for finding EF1 allocations in the following settings: (i) n agents with heterogeneous partition matroids and heterogeneous binary valuations, (ii) 2 agents with heterogeneous partition matroids and heterogeneous valuations, and (iii) at most 3 agents with heterogeneous binary valuations and identical base-orderable matroids.

1 Introduction

Many real-life problems involve the fair allocation of indivisible items among agents with different preferences, and with constraints on the bundle that each agent may receive. Examples include the allocation of course seats among students (Budish et al. 2017) and the allocation of conference papers among referees (Garg et al. 2010).

In general, different agents may have different constraints. For example, consider the allocation of employees among departments of a company: one department has room for four project managers and two backend engineers, while another department may have room for three backend engineers and five data scientists. Another example can be found in the way shifts are assigned among medical doctors, where every doctor has her own schedule limitations.

Our goal is to devise algorithms that find fair allocations of indivisible items among agents with different preferences and different feasibility constraints. Let us first explain what we mean by "constraints" and what we mean by "fair".

We focus on constraints that are represented by *matroids* (see Section **2.1**), mainly *partition matroids*. In a partition matroid, the set of items is partitioned into a set of *categories*, and every category is associated with a cap on the number of items from that category that can be allocated to each agent.

A classic notion of fairness is *envy freeness* (EF), which means that every agent (weakly) prefers his or her bundle to that of any other agent. Since an EF allocation may not exist when items are indivisible, recent studies focus on its relaxation known as *EF1* — envy free up to one item (Budish 2011) — which means that every agent i (weakly) prefers her bundle to any other agent j's bundle, up to the removal of the best good (in i's eyes) from agent j's bundle (see Section **2.2**).

Without feasibility constraints, an EF1 allocation always exists and can be computed efficiently (Lipton et al. 2004). However, this result does not consider feasibility constraints. There are two ways to address such constraints.

The first approach is to directly construct allocations that satisfy the constraints, i.e., guarantee that each agent receives a feasible bundle. This approach was taken recently by Biswas and Barman (2018, 2019), who study settings with *additive valuations*, where every agent values each bundle at the sum of the values of its items. They present efficient algorithms for computing EF1 allocations when agents have: (i) identical matroid constraints and identical valuations; or (ii) identical partition matroid constraints, even under heterogeneous valuations (see Section **2.3**). However, their algorithms do not handle different partition constraints, or identical matroid constraints with different valuations.

A second approach is to capture the constraints within the valuation function. That is, the value of an agent for a bundle equals the value of the best feasible subset of that bundle. This approach seamlessly addresses heterogeneity in both constraints and valuations. The valuation functions constructed this way are no longer additive, but are *submodular* (Oxley 2006). Recently, Babaioff, Ezra, and Feige (2020)

*For the full version, see https://arxiv.org/abs/2010.07280 (Dror, Feldman, and Segal-Halevi 2020). Due to space limitation, all missing statements and proofs are deferred to the full version.

and Benabbou et al. (2020) have independently proved the existence of EF1 allocations in the special case in which agents have submodular valuations with *binary marginals* (where adding an item to a bundle adds either 0 or 1 to its value). Such an allocation can be converted to a fair *and feasible* allocation by giving each agent the best feasible subset of his/her allocated bundle, and disposing of the other items.

However, in some settings, such disposal of items may be impossible. For example, when allocating shifts to medical doctors, if an allocation rule returns an infeasible allocation and shifts are disposed to make it feasible, the emergency-room might remain understaffed. A similar problem may occur when allocating papers to referees, where disposals may leave some papers without reviews. The allocation rules developed in the above papers may not yield EF1 allocations when they are constrained to return feasible allocations. Thus, an open problem remains:

Open problem*: Given agents with* different *additive valuations and* different *matroid constraints, which settings admit a* complete *and* feasible *EF1 allocation?*

1.1 Contribution and Techniques

Feasible envy. Before presenting our results, we shall discuss the EF1 notion in settings with heterogeneous constraints. Consider a setting with two agents, Alice and Bob, and 8 identical items of a single category, with capacities 3 and 5 for Alice and Bob, respectively. Every complete feasible allocation gives 3 items to Alice and 5 to Bob. Ignoring feasibility constraints, such an allocation is not EF1, since even after removing a single item from Bob's bundle, Alice values it at 4, which is greater than her value for her own bundle. However, a bundle of 4 items is infeasible for Alice. Therefore, a more reasonable definition of envy in this setting is *feasible envy*, where each agent compares her bundle to the best feasible subset of any other agent's bundle (see Section 2.2 for formal definition). In the example above, the best feasible subset of Bob's bundle for Alice is worth 3. Thus, the allocation is *feasibly-envy-free* (F-EF).

If Alice values one of Bob's items at 2, then the above allocation is not F-EF, since the best feasible subset of Bob's bundle for Alice is worth 4, but it is *F-EF1*, as it becomes F-EF after removing this item from Bob's bundle. Throughout the paper, we use the notion of F-EF1 under heterogeneous constraints. Note that F-EF1 is equivalent to EF1 when agents have identical constraints.

Impossibilities. Below, we present several impossibility results that direct us to the interesting domain of study.

First, if the partition of items into categories is different for different agents, an F-EF1 allocation may not exist, even for two agents with identical valuations.

Second, going beyond matroid constraints to *graph-matching constraints* (an intersection of two matroids) is futile: even with two agents with identical valuations and identical matching constraints, an F-EF1 allocation may not exist.

Third, going beyond EF1 to the stronger notion of *envy-free up to any good (EFX)* is futile: even with two agents with identical valuations and identical uniform matroid constraints, an EFX allocation may not exist.

Based on these results, we focus on finding F-EF1 allocations when the agents' constraints are represented by either: (1) *partition matroids* where all agents share the same partition of items into categories but may have different capacities; or (2) *base-orderable (BO) matroids* — a wide class of matroids containing partition matroids — where all agents have identical matroid constraints but possibly different valuations.

Algorithms (see Table 1). For *partition matroids*, the reason that the algorithms of Lipton et al. (2004) and Biswas and Barman (2018) fail for agents with different capacities is that they rely on *cycle removal* in the envy graph. Informally (see Section 2.3 for details), these algorithms maintain a directed *envy graph* in which each agent points to every agent she envies. The algorithm prioritizes the agents who are not envied, since giving an item to such agents keeps the allocation EF1. If there are no unenvied agents, the envy graph must contain a cycle, which is then removed by exchanging bundles along the cycle. However, when different agents in the cycle have different constraints, this exchange may not be feasible. Thus, our main challenge is to develop techniques that guarantee that no envy-cycles are created in the first place. We manage to do so in four settings of interest:

1. There are at most *two categories* (see Section **3**).
2. All agents have *identical* valuations (see Section **3**).
3. All agents have *binary valuations* (see Section **4**).
4. There are *2 agents* (see Section **5**).

Each setting is addressed by a different algorithm and using a different cycle-prevention technique.

Beyond partition matroids, we consider the much wider class of matroids, termed *base-orderable* (BO) matroids (see definition 6.1). This class contains partition matroids, laminar matroids (an extension of partition matroids where the items in each category can be partitioned into subcategories), transversal matroids, and other interesting matroid structures. In fact, it is conjectured by Bonin and Savitsky (2016) that "almost all matroids are BO". For this class we present algorithms for agents with identical constraints and different additive valuations in the following cases:

5. There are *2 agents* (see full version).
6. There are *3 agents* with *binary valuations* (see Section**6**).

All of our algorithms run in polynomial-time.

1.2 Related Work

Capacity constraints are common in matching markets such as doctors–hospitals and workers–firms: see Klaus, Manlove, and Rossi (2016) for a recent survey. In these settings, the preferences are usually represented by ordinal rankings rather than by utility functions, and the common design goals are Pareto efficiency, stability and strategy-proofness rather than fairness.

Fair allocation with capacity constraints is particularly relevant to the problem of *assigning conference papers to referees*. Garg et al. (2010); Long et al. (2013); Lian et al.

Matroid Type	Complete allocation	Het. constraints	Het. valuations	Valuations	# of Agents	Remarks	Reference
Partition	✓	-	✓	General	n		Biswas and Barman (2018)
	✓	✓	-	General	n		Section 3
	✓	✓	✓	General	2		Section 5
	✓	✓	✓	General	n	$\leq$ 2 categories	Section 3
	✓	✓	✓	Binary	n	PE for binary caps	Section 4 / Full Ver.
Beyond Partition	✓	-	-	General	n	Laminar	Biswas and Barman (2018)
	-	✓	✓	Binary	n	PE General	Babaioff, Ezra, and Feige (2020) Benabbou et al. (2020) ***
	✓	-	✓	Binary	3	PE BO	Section 6
	✓	-	✓	General	2	BO	Section 6

Table 1: A summary of the results. All results are for additive valuations. PE = Pareto-efficient. BO = base-orderable (Sec. 6). The line marked by *** is not mentioned explicitly, but follows directly from their results (Sec. 1).

(2018) all study a setting in which there is both an upper and a lower capacity on the total number of items for each agent (reviewer). The constraints may be different for each agent, but there is only one category of items. Note that lower capacities are not matroid constraints, since they are not downwards-closed. The same is true in the setting studied by Ferraioli, Gourvès, and Monnot (2014), where each agent must receive exactly k items.

Fair allocation of items of different categories has been studied by Mackin and Xia (2016) and Sikdar, Adali, and Xia (2017). There are k categories, each of which has n items, and each agent must receive exactly one item of each category. Sikdar, Adalı, and Xia (2019) consider an exchange market where each agent holds multiple items of each category and should receive a bundle with exactly the same number of items of each category. Nyman, Su, and Zerbib (2020) study a similar setting, but with monetary transfers.

Barrera et al. (2015), Bilò et al. (2018), and Suksompong (2019) study another kind of constraint in fair allocation. The goods are arranged on a line, and each agent must receive a connected subset of the line, as when each item is a house, and each agent should get a connected part of the street. Bouveret et al. (2017) ant Bei et al. (2019) study a more general setting in which the goods are arranged on a general graph, and each agent must receive a connected subgraph. Note that these are not matroid constraints.

Gourvès, Monnot, and Tlilane (2013) study a setting with a single matroid, where the goal includes building a base of the matroid and providing worst case guarantees on the agents' utilities. Gourvès, Monnot, and Tlilane (2014) and Gourvès and Monnot (2019) require the *union* of bundles allocated to all agents to be an independent set of the matroid. This inherently requires to leave some items unallocated, which we do not allow here.

Fair allocation with *binary additive valuations* (without constraints) has been studied recently, due to its practical applications (Aleksandrov et al. 2015). With binary valuations, better fairness guarantees (Bouveret and Lemaître 2016; Barman et al. 2018; Amanatidis et al. 2020) and better strategic properties (Halpern et al. 2020) can be attained. While, in general, the MNW solution is NP-hard, with binary valuations it can be computed efficiently (Darmann and Schauer 2015; Barman, Krishnamurthy, and Vaish 2018).

2 Model and Preliminaries

2.1 Allocations and Constraints

We consider settings where a set M of m items should be allocated among a set N of n agents. An allocation is denoted by $\mathbf{X} = (X_1, \dots, X_n)$, where $X_i \subseteq M$ is the bundle given to agent i, and $X_i \cap X_j = \emptyset$ for all $i \neq j \in N$. An allocation is *complete* if $\biguplus_{i \in N} X_i = M$. Throughout, we use $[n]$ to denote the set $\{1, \dots, n\}$.

We consider constrained settings, where every agent i is associated with a matroid $\mathcal{M}_i = (M, \mathcal{I}_i)$ that specifies the feasible bundles for agent i.

Definition 2.1. Given the set of items M, and a nonempty set of *independent sets* $\mathcal{I} \subseteq 2^M$, the pair $\mathcal{M} = (M, \mathcal{I})$ is a *matroid* if it satisfies the following properties: (i) *Downward-closed*: for every $S, S' \subseteq S$, if $S \in \mathcal{I}$, then $S' \in \mathcal{I}$; and (ii) For every $S, T \in \mathcal{I}$, if $|S| > |T|$, then there exists $g \in S \setminus T$ such that $T \cup \{g\} \in \mathcal{I}$. A *base* of a matroid $\mathcal{M}$ is a maximal cardinality independent set in $\mathcal{M}$.

A special case of a matroid is a *partition matroid*:

Definition 2.2. (partition matroid) A matroid $\mathcal{M}_i = (M, \mathcal{I}_i)$ is a *partition matroid* if: (i) The set of items M is partitioned into a set of categories $C_i = \{C_i^1, \dots, C_i^{\ell_i}\}$ for some $\ell_i \leq m$, (ii) Categories are associated with *capacities* $k_i^1, \dots, k_i^{\ell_i}$, and (iii) The collection of independent sets is $\mathcal{I}_i = \{S \subseteq M : |S \cap C_i^h| \leq k_i^h \text{ for every } h \in [\ell_i]\}$.

Given an allocation $\mathbf{X}$, we denote by $\mathbf{X}_i^h$ the items from category C_i^h given to agent i in $\mathbf{X}$. A special case of a partition matroid is a *uniform matroid*, which is a partition matroid with a single category.

Definition 2.3. (feasible allocation) An allocation $\mathbf{X}$ is said to be *feasible* if: (i) it is individually feasible: $X_i \in \mathcal{I}_i$ for every agent i, and (ii) it is complete: $\biguplus_i X_i = M$.

Let $\mathcal{F}$ denote the set of all feasible allocations. Throughout this paper we consider only instances that admit a feasible allocation:

Assumption 2.4. *All instances considered in this paper admit a feasible allocation; i.e., $\mathcal{F} \neq \emptyset$. For partition matroids, feasibility means that for every category C^h, the sum of agent capacities for this category is at least $|C^h|$.*

An instance is said to have *identical matroids* if all agents have the same matroid feasibility constraints. I.e., $\mathcal{I}_i = \mathcal{I}_j$ for all $i, j \in N$. An instance with partition matroids is said to have *identical categories* if all the agents have the same partition into categories. I.e., $\ell_i = \ell_j = \ell$ for every $i, j \in N$, and $C_i^h = C_j^h = C^h$ for every $h \in \ell$. The capacities, however, may be different.

2.2 Valuations and Fairness Notions

Every agent i is associated with an *additive* valuation function $v_i : 2^M \to R^+$, which assigns a positive real value to every set $S \subseteq M$. Additivity means that there exist m values $v_i(1), \ldots, v_i(m)$ such that $v_i(S) = \sum_{j \in S} v_i(j)$. An additive valuation v_i is called *binary* if $v_i(j) \in \{0, 1\}$ for every $i \in N, j \in M$. An allocation $\mathbf{X}$ is *Social Welfare Maximazing* (SWM) if $\mathbf{X} = \text{argmax}_{\mathbf{X}' \in \mathcal{F}} \sum_{i \in [n]} v_i(X_i')$.

Definition 2.5 (envy and envy freeness)**.** Given an allocation $\mathbf{X}$, agent i **envies** agent j iff $v_i(X_i) < v_i(X_j)$. $\mathbf{X}$ is *envy free* iff no agent envies another agent.

Definition 2.6 (EF1)**.** (Budish 2011) An allocation $\mathbf{X}$ is *envy free up to one good (EF1)* iff for every $i, j \in N$, there exists a subset $Y \subseteq X_j$ with $|Y| \leq 1$, such that $v_i(X_i) \geq v_i(X_j \setminus Y)$.

Definition 2.7. The *best feasible subset* of a set S for agent i is $\text{BEST}_i(S) := \text{argmax}_{T \subseteq S,\, T \in \mathcal{I}_i} v_i(T)$. While $\text{BEST}_i(S)$ is not necessarily unique, we abuse notation and use $\text{BEST}_i(S)$ as an arbitrary set in $\text{argmax}_{T \subseteq S,\, T \in \mathcal{I}_i} v_i(T)$.

Definition 2.8 (feasible valuation)**.** The *feasible valuation* of agent i for a set S is $\hat{v}_i(S) := v_i(\text{BEST}_i(S))$.

Definition 2.9. Given a feasible allocation $\mathbf{X}$:

- Agent i *F-envies* agent j iff $\hat{v}_i(X_i) < \hat{v}_i(X_j)$.
- $\mathbf{X}$ is *F-EF* (feasible-EF) if no agent F-envies another one.
- $\mathbf{X}$ is *F-EF1* iff for every $i, j \in N$: there exists a subset $Y \subseteq X_j$ with $|Y| \leq 1$, such that $\hat{v}_i(X_i) \geq \hat{v}_i(X_j \setminus Y)$.

For further discussion of the F-EF1 criterion, and an alternative (weaker) definition, see the full version.

Another usfull notation is *positive feasible envy*, which is the amount by which an agent F-envies another agent:

Definition 2.10. The *positive feasible envy* of agent i towards j in allocation $\mathbf{X}$ is:
$Envy^+_{\mathbf{X}}(i, j) := \max(\hat{v}_i(X_j) - \hat{v}_i(X_i), 0)$.

Definition 2.11. The *envy graph* of an allocation $\mathbf{X}$, $\mathcal{G}(\mathbf{X})$, is a directed graph where the nodes represent the agents, and there is an edge from agent i to agent j iff $v_i(X_i) < v_i(X_j)$. We use the term *feasible envy graph* to indicate an envy graph created by the feasible-envy instead of plain envy.

2.3 Common Tools and Techniques

Below we review the most common methods for finding an EF1 allocation.

Envy cycle elimination The first method for attaining an EF1 allocation (in unconstrained setting, even with arbitrary valuations) is due to Lipton et al. (2004).

The *envy cycles elimination* algorithm works as follows: it starts with the empty allocation. Then, as long as there is an unallocated item: (i) choose an agent that is a source in the envy graph (i.e., no agent envies her), and give her an arbitrary unallocated item, (ii) reconstruct the envy graph $\mathcal{G}$ corresponding to the new allocation, (iii) as long as $\mathcal{G}$ contains cycles, choose an arbitrary cycle, and shift the bundles along the cycle. This increases the total value, thus this process must end with a cycle-free graph.

Max Nash welfare The *Nash social welfare* (NW) of an allocation $\mathbf{X}$ is the geometric mean of the agents' values: $NW = (\prod_{i \in [n]} v_i(X_i))^{\frac{1}{n}}$. An allocation is *max Nash welfare (MNW)* if it maximizes the NW among all feasible allocations. Caragiannis et al. (2019) showed that in unconstrained settings with additive valuations, every MNW allocation is EF1.

Round robin (RR) RR works as follows: given a fixed order σ over the agents, as long as there is an unallocated item, the next agent according to σ (where the next agent of agent n is agent 1) chooses an item she values most among the unallocated items. Simple as it might be, this algorithm results in an EF1 allocation in unconstrained settings with additive valuations (Caragiannis et al. 2019)

Per category RR + envy cycle elimination This algorithm was introduced by Biswas and Barman (2018) for finding an EF1 allocation in settings with homogeneous partition constraints. It resolves the categories sequentially, resolving each one by RR followed by envy cycle elimination, where the order over the agents is determined by a topological order in the obtained envy graph.

3 Warmup: Uniform Matroids

As a warm-up, we present a simple algorithm termed Capped Round Robin (CRR). CRR is a slight modification of round robin, where if an agent reached her capacity — she is being skipped over; CRR finds a F-EF1 allocation whenever the constraints of all agents are *uniform matroids*, i.e., all items belong to a single category (but agents may have different capacities and different valuations).

While CRR may not find an F-EF1 allocation for more than one category, we can extend it to two categories by running CRR with reverse order on the second category.

Theorem 3.1. *When all agents have partition-matroid constraints with at most two categories, the same categories but possibly different capacities, an F-EF1 allocation always exists and can be found efficiently.*

Using CRR as a subroutine, we show that a similar algorithm to the one used by Biswas and Barman (2018) finds an F-EF1 allocation in settings with partition matroids with different capacities and identical valuations; this follows from the fact that no cycles can be formed in the envy graph.

Theorem 3.2. *When all agents have partition-matroid constraints with the same categories but possibly different capacities, and identical additive valuations, an F-EF1 allocation always exists and can be found efficiently.*

4 Partition Matroids with Binary Valuations

In this section we present an algorithm that finds an F-EF1 allocation in settings with n agents with different binary valuations, and partition matroids with different capacity constraints. For binary valuations $v_i(j) \in \{0, 1\}$ for all i, j, and for every agent i we refer to the set of items $J = \{j \in M \ s.t \ v_i(j) = 1\}$ as agent i's *desired set*.

Theorem 4.1. *In every setting with partition matroids with binary valuations (possibly heterogeneous capacities and heterogeneous valuations), an F-EF1 allocation exists and can be computed efficiently by the* Iterated Priority Matching Algorithm *(described below).*

A key tool we use is *priority matching*, defined next.

Priority matching. Given a graph $G = (V, E)$, a *matching* in G is a subset of edges $\mu \subseteq E$ such that each vertex $u \in V$ is adjacent to at most one edge in μ. Given an ordering on the vertices, $\sigma[1], \dots, \sigma[n]$, every matching is associated with a binary vector of size n, where element i equals 1 whenever vertex $\sigma[i]$ is matched. The priority matching is the matching associated with the maximum vector in the lexicographic order. Note that every ordering over the vertices potentially yields a different priority-matching.

Priority matching was introduced by Roth, Sönmez, and Ünver (2005) in the context of kidney exchange, where they prove that every priority matching is also a *maximum-cardinality matching*; that is, it maximizes the total number of saturated vertices in V.[1]

We next describe the algorithm.

Algorithm *Iterated Priority Matching*. (for pseudo code see full version.) The algorithm works category-by-category. For each category h, the items of C^h are allocated in two phases, namely the *matching phase* and the *leftover phase*. The matching phase proceeds in several iterations, where in each iteration, every agent receives at most one item. The number of iterations is at most the maximum capacity of an agent in C^h, denoted by $T^h := \max_{i \in N} k_i^h$. In each iteration t of the matching phase, we construct a bipartite graph G_t^h, where one side consists of the agents with remaining capacity (i.e., agents such that $|X_i^h| < k_i^h$), and one side consists of the unallocated items of C^h. An edge (i, j) exists in G_t^h iff j is a desired item of i (i.e., $v_i(j) = 1$). Given the current allocation, let σ be a topological order over the agents in the feasible envy graph (we shall soon show that the feasible envy-graph is cycle free). We compute a *priority matching* in G_t^h with respect to σ, and augment agent allocations by the obtained priority matching. We then update the feasible envy graph and proceed to the next iteration, where the next set of items in C^h is allocated.

After at most T^h iterations, all remaining items of category C^h contribute value 0 to all agents with remaining capacity, and we move to the *leftover phase*. In this phase, we allocate the leftover items arbitrarily among agents, respecting feasibility constraints. This is possible since a feasible allocation exists by assumption.

To prove the correctness of the algorithm, it suffices to prove that every feasible envy-graph constructed in the process is cycle-free, and that the feasible envy between any two agents is at most 1. We prove both conditions simultaneously in the following theorem.

Lemma 4.2. *In every iteration of the algorithm:*
(a) The feasible envy-graph has no cycles;
(b) For every $i, j \in N$, $Envy^+_{\mathbf{X}}(i, j) \leq 1$.

Proof. The proof is by induction on the categories and iterations (details below). Both claims clearly hold from the outset (i.e., under the empty allocation). In our analysis we refer to states before and after (h, t) to denote the states before and after iteration t of category h, respectively.

We start by proving property (a): Assume that property (a) holds before $(h, 1)$ (i.e., before starting to allocate items in category h). We prove that it holds after (h, t) for every t. Suppose by contradiction that after (h, t) there is a cycle $i_1 \to \cdots \to i_p = i_1$ in the feasible envy-graph. By assumption (a) the cycle did not exist before category h, so at least one edge was created during the first t steps in category h. Suppose w.l.o.g. that it is the edge $i_1 \to i_2$.

Let Q_1 be the set of items desired by i_1 that are allocated to i_1 up to iteration t of category h, and let $q = |Q_1|$. It must hold that i_1 got these q items in the first q iterations of h (otherwise, there exists an iteration $\leq q$ in which i_1 did not get an item, but a desired item remained unallocated, contradicting maximum priority matching). Let Q_2 be the set of items desired by i_1 that are allocated to i_2 up to iteration t of category h. The fact that i_1 started to envy i_2 during category h implies that $|Q_2| \geq q + 1$ and $k_{i_1}^h \geq q + 1$. It must hold that i_2 got these $q + 1$ items in the first $q + 1$ iterations of h (otherwise, one of these items could be allocated to i_1 in iteration $q + 1$, contradicting maximum priority matching). This also implies that iteration $q + 1$ is still within the matching phase, since there is an item desired by i_1, and i_1 has remaining capacity. Therefore, i_2 received at least $q + 1$ items within the matching phase, implying that i_2's value increased by at least $q + 1$ up to iteration t of category h.

Let Q_3 be the set of items desired by i_2 that are allocated to i_3 up to iteration t of category h. i_2 envies i_3 after (h, t), and before $(h, 1)$ $Envy^+_{\mathbf{X}}(i_2, i_3) \leq 1$. Since i_2's value increased by at least $q + 1$ up to iteration t of category h, it must hold that $|Q_3| \geq q + 1$. We now claim that before $(h, q + 1)$, at most one item of Q_3 was available, and i_3 got it in this iteration. Otherwise, one could allocate one of those items to i_2, and allocate the item that i_2 received in iteration $q + 1$ (that is desired by i_1), to i_1, increasing the priority matching.

We conclude that i_3 got an item at each one of the first $q + 1$ iterations of category h, as $|Q_3| \geq q + 1$. Since all of

[1] Okumura (2014) extends this result to priority classes of arbitrary sizes, and shows a poly-time algorithm for finding a priority matching. Simpler algorithms were presented by Turner (2015b,a).

these iterations are within the matching phase, all of these items are desired by i_3. Therefore, i_3's value increases by at least $q+1$. Repeating this argument, we conclude that every agent along the cycle received at least $q+1$ desired items during the first t steps of h, including agent $i_p = i_1$; but this is in contradiction to the fact that i_1 received $q = |Q_1|$ items.

We next prove property (b): We assume that property (b) holds for every iteration before (h,t) and prove that it holds after (h,t). Suppose by contradiction that after (h,t) $Envy^+_{\mathbf{X}}(i,i') > 1$ for some agents i,i'. From the induction assumption, before (h,t), $Envy^+_{\mathbf{X}}(i,i') \leq 1$, and since at most one item can be allocated to i' in iteration t, we conclude that before (h,t), $Envy^+_{\mathbf{X}}(i,i') = 1$. Therefore, i precedes i' in the topological order σ in iteration t. Since the envy of i towards i' increased, the algorithm must have allocated to i' some item $j \in C^h$ desired by i in iteration t.

We distinguish between two cases. Case (1): The capacity of agent i was not exhausted before (h,t). Then, the priority-matching on G^h_t would prefer the matching in which j is given to i over the one where j is given to i'; a contradiction. Case (2): The capacity of agent i was exhausted before (h,t). Then, since j is an available item that i desires, it must be that the capacity of i was exhausted during the matching phase. This implies that i desires all the items she receives in category C^h. That is, if $\mathbf{X}$ is the allocation after (h,t), $v_i(X^h_i) = k^h_i \geq \hat{v}_i(X^h_{i'})$. By the contradiction assumption, after (h,t), $Envy^+_{\mathbf{X}}(i,i') > 1$; that is, $v_i(X_i) \leq \hat{v}_i(X_{i'})-2$. Let $\mathbf{X}'$ be the allocation before $(h,1)$. By additivity and the above inequalities, we get that $v_i(X'_i) = v_i(X_i) - v_i(X^h_i) \leq \hat{v}_i(X_{i'}) - 2 - \hat{v}_i(X^h_{i'}) = \hat{v}_i(X'_{i'}) - 2$, implying that the allocation was not F-EF1 before category h; a contradiction. □

5 Partition Matroids with Two Agents

In this section we present an algorithm for 2 agents.

Theorem 5.1. *In every setting with two agents and partition matroid constraints, an F-EF1 allocation exists and can be computed efficiently by RR^2.*

To present the algorithm we introduce some notation.

- Given an allocation $\mathbf{X}$, the *surplus* of agent i in category h is $s^h_i(\mathbf{X}) := \hat{v}_i(X^h_i) - \hat{v}_i(X^h_j)$.
  I.e., it is the difference between agent i's value for her own bundle and her value for agent j's bundle.
- Given agents $1, 2$, $\ell \in \{1,2\}$, valuation functions v, v' and category h, $\chi(v,v',\ell)^h$ is the allocation obtained by Capped Round Robin (CRR) (see Section 2) for category h, under valuations $v_1 = v, v_2 = v'$, and where agent ℓ plays first. When clear in the context, we omit the superscript h from $\chi(v,v',\ell)^h$.

We are now ready to present Algorithm "Round Robin Squared" (RR^2) . In RR^2, there are two layers of round robin (RR), one layer for choosing the next category, and one layer for choosing items within a category. For every agent i, the categories are ordered based on $s^h_i(\chi(v_1,v_2,i))$, in a non-increasing order; call this order π_i. In the first iteration, agent 1 chooses the first category in π_1. Within this category, the items are allocated according to CRR, with agent 1 choosing first. In the second iteration, agent 2 chooses the first category in π_2 that has not been chosen yet. Within this category, the items are allocated according to CRR, with agent 2 choosing first. The algorithm proceeds in this way, where in every iteration, the agent who chooses the next category flips; that agent chooses the highest unallocated category in her surplus-order, and within that category, agents are allocated according to CRR with that agent choosing first. This proceeds until all categories are allocated.

The key lemma in our proof asserts that the surplus of an agent i when playing first within a category h is at least as large as minus the surplus of the same agent when playing second in the same category. I.e.,

Lemma 5.2. *For every category h and every $i = 1,2$: $s^h_i(\chi(v_1,v_2,i)^h) \geq -s^h_i(\chi(v_1,v_2,j)^h)$, where $j = 3-i$.*

We now show how Lemma 5.2 implies the assertion of Theorem 5.1.

Proof of Theorem 5.1. We first show that the first agent choosing a category does not F-envy the other agents. That is, $v_i(X_i) \geq \hat{v}_i(X_j)$, where $\mathbf{X}$ is the outcome of RR^2 and i is the first agent to choose. By reordering, let $C^1, ..., C^\ell$ be the categories in the order they are chosen, and let agent 1 choose a category first. It holds that:

$$v_1(X_1) - \hat{v}_1(X_2) = \sum_{h=1,\ldots,\ell} v_1(X^h_1) - \sum_{h=1,\ldots,\ell} \hat{v}_1(X^h_2)$$

$$= \sum_{h=1,\ldots,\ell} (v_1(X^h_1) - \hat{v}_1(X^h_2))$$

$$= \sum_{h \text{ is odd}} s^h_1(\chi(v_1,v_2,1)) + \sum_{h \text{ is even}} s^h_1(\chi(v_1,v_2,2)) \tag{1}$$

$$\geq \sum_{h \text{ is odd}} s^h_1(\chi(v_1,v_2,1)) + \sum_{h \text{ is even}} -s^h_1(\chi(v_1,v_2,1)) \tag{2}$$

$$= \sum_{t=1,\ldots,\frac{\ell}{2}} (s^{2t-1}_1(\chi(v_1,v_2,1)) - s^{2t}_1(\chi(v_1,v_2,1))). \tag{3}$$

The first equations follow from additivity. Equation 1 follows from the definition of surplus, the facts that agent 1 chooses the odd categories, and the agent who chooses the category is the one to choose first within this category. Inequality 2 follows from Lemma 5.2. If the number of categories is odd, we add a dummy empty category.

Since agent 1 chooses the odd categories, and she does so based on highest surplus, for every t $s^{2t-1}_1(\chi(v_1,v_2,1)) \geq s^{2t}_1(\chi(v_1,v_2,1))$, as category $2t$ was available when agent 1 chose category $2t-1$. Therefore, every summand in the sum of (3) is non-negative. Thus, the whole sum is non-negative, implying that $v_1(X_1) \geq \hat{v}_1(X_2)$, as desired.

We next show that agent 2 does not F-envy agent 1 beyond F-EF1. As a thought experiment, consider the same setting with the first chosen category removed. Following the same reasoning as above, in this setting agent 2 does not F-envy agent 1. But within the first category, agent 2 can only F-envy agent 1 up to 1 item. That is, there exists one item in the first category such that when it is removed, it eliminates the feasible envy of the second agent within that category, and thus eliminates her feasible envy altogether. We conclude that the obtained allocation is F-EF1. □

6 BO Matroids with up to Three Agents

In this section we consider constraints that are represented by a wide class of matroids, termed *base-orderable* (BO) matroids, defined as follows:

Definition 6.1 (Brualdi and Scrimger (1968))**.** A matroid is BO if for every two bases I, J, there exists a bijection $\mu : I \to J$ such that for any $i \in I$: $I \setminus \{i\} \cup \{\mu(i)\} \in \mathcal{I}$ and $J \setminus \{\mu(i)\} \cup \{i\} \in \mathcal{I}$.

This class contains many interesting matroids, including partition matroids, laminar matroids, transversal matroids, and more. Bonin and Savitsky (2016) conjectures that almost all matroids are BO. When different agents have different matroids, an F-EF1 allocation may not exist. Therefore, we restrict attention to settings with identical matroids. In this setting, Biswas and Barman (2019) find an EF1 allocation for n agents with laminar matroids and *identical* valuations. Using their algorithm as a subroutine, and extending it to BO matroids, it is easy to find an EF1 allocation for $n = 2$ agents with *different* valuations (see full version). The case of $n \geq 3$ heterogeneous additive agents remains open. In what follows, we establish existence for 3 agents with heterogeneous *binary* valuations.

Theorem 6.2. *For identical BO matroid constraints, for 3 agents with heterogeneous binary valuations, there exists an EF1 allocation that is also social welfare maximizing(SWM) (hence Pareto-efficient).*

The proof uses an algorithm, called *Iterated Swaps* (see full version). The algorithm starts by finding an SWM allocation $\mathbf{X}$. Given such allocation, the algorithm constructs a new allocation $\hat{X}$ which is EF1 and SWM. The algorithm works as follows: as long as there exist agents i, j such that $Envy^+_{\mathbf{X}}(i, j) > 1$, if possible (feasibility-wise), transfer an item desired by i from j to i. Otherwise, swap items between i and j, such that i gets from j an item that i desires, and j gets from i an item that i does not desire. We prove that one of these options always exists, and the process terminates with an EF1 and SWM allocation.

Definition 6.3. Given a matroid $(M, \mathcal{I})$ and 2 independent sets $I, J \in \mathcal{I}$, items $a \in I$ and $b \in J$ represent a *feasible swap* if both $(J \setminus \{b\}) \cup \{a\}$ and $(I \setminus \{a\}) \cup \{b\}$ are in $\mathcal{I}$.

To prove Theorem 6.2 we use the following lemmas.

Lemma 6.4. *In a BO matroid-constrained setting with binary valuations, if agent i envies agent j under a feasible SWM allocation $\mathbf{X}$, then one of the following holds:*

1. *There exists an item $a \in X_j$ s.t. $v_i(a) = v_j(a) = 1$ and $X_i \cup \{a\} \in \mathcal{I}$;*
2. *There exist items $a \in X_j$, $b \in X_i$ s.t. $v_i(a) = v_j(a) = 1$, $v_i(b) = v_j(b) = 0$, and a, b represent a feasible swap.*

We refer to such a transfer or swap as a *smart move*. The following is a direct corollary of Lemma 6.4:

Corollary 6.5. *Given a SWM allocation $\mathbf{X}$, an allocation $\mathbf{X}'$ obtained from $\mathbf{X}$ by a smart move is also SWM.*

Finally, we also use the following lemma:

Lemma 6.6. *Let $\mathbf{X}$ be a SWM allocation where $Envy^+_{\mathbf{X}}(i, j) > 1$, and let $\mathbf{X}'$ be an allocation obtained from $\mathbf{X}$ by a smart move. Then:*
(i) $Envy^+_{\mathbf{X}'}(i, j) = Envy^+_{\mathbf{X}}(i, j) - 2$; (ii) $Envy^+_{\mathbf{X}'}(j, i) = 0$.

We are now ready to prove Theorem 6.2. In the proof, when we mention a change in $Envy^+_{\mathbf{X}}(i, j)$ we refer to the change in the positive envy of agent i to agent j between allocations $\mathbf{X}$ and $\mathbf{X}'$; i.e., to $Envy^+_{X'}(i, j) - Envy^+_{\mathbf{X}}(i, j)$.

Proof of Theorem 6.2. Let $\mathbf{X}$ be a feasible SWM allocation. If $\mathbf{X}$ is EF1, we are done. Otherwise, as long as $\mathbf{X}$ is not EF1, choose some pair of agents i, j such that $Envy^+_{\mathbf{X}}(i, j) > 1$. Since $\mathbf{X}$ is SWM (by Corollary 6.5), j does not envy i (otherwise, switching i and j's bundles increases social welfare, a contradiction to SWM).

Let $\Phi(\cdot)$ be the following potential function:
$\Phi(\mathbf{X}) := \sum_i \sum_{j \neq i} Envy^+_{\mathbf{X}}(i, j)$. By Lemmas 6.4 and 6.6, there must exist a feasible smart move between i, j such that the social welfare remains unchanged, $Envy^+_{\mathbf{X}}(i, j)$ drops by 2 and $Envy^+_{\mathbf{X}}(j, i)$ remains 0. Thus, $Envy^+_{\mathbf{X}}(i, j) + Envy^+_{\mathbf{X}}(j, i)$, drops by 2.

Let us next consider the positive envy that might be added due to terms of Φ that include the third agent, deonte it by k.

1. $Envy^+_{\mathbf{X}}(i, k)$ cannot increase, as the smart move improves i's valuation, while $v_i(X_k)$ does not change.
2. $Envy^+_{\mathbf{X}}(k, i)$ increases by at most 1: the largest possible increase in $v_k(X'_i)$ is 1, while $v_k(X_k)$ does not change.
3. $Envy^+_{\mathbf{X}}(k, j)$ increases by at most 1: the largest possible increase in $v_k(X'_j)$ is 1, while $v_k(X_k)$ does not change.
4. $Envy^+_{\mathbf{X}}(j, k)$ increases by at most 1, as this is the exact decrease in $v_j(X'_j)$, while $v_j(X_k)$ does not change.

We next claim that among the terms that may increase by 1 (#2, #3, #4), no two of them can increase simultaneously:

- $Envy^+_{\mathbf{X}}(k, j), Envy^+_{\mathbf{X}}(j, k)$ cannot increase simultaneously as this would create an envy-cycle, contradicting SWM.
- $Envy^+_{\mathbf{X}}(k, i), Envy^+_{\mathbf{X}}(j, k)$ cannot increase simultaneously, as this together with the fact that $Envy^+_{\mathbf{X}}(i, j) \geq 0$ contradicts SWM. Indeed shifting bundles along the cycle $i \to j \to k \to i$ strictly increases welfare.
- $Envy^+_{\mathbf{X}}(k, i), Envy^+_{\mathbf{X}}(k, j)$ cannot increase simultaneously as the sum of k's values to i and j's bundles is fixed; i.e., $v_k(X_i) + v_k(X_j) = v_k(X'_i) + v_k(X'_j)$.

We conclude that in every iteration the potential function drops by at least 1. Indeed, $Envy^+_{\mathbf{X}}(i, j)$ drops by 2, $Envy^+_{\mathbf{X}}(j, i)$ remains 0, $Envy^+_{\mathbf{X}}(i, k)$ does not change, and among $Envy^+_{\mathbf{X}}(k, i), Envy^+_{\mathbf{X}}(k, j), Envy^+_{\mathbf{X}}(j, k)$ only one can increase, by at most 1. Since Φ is lower bounded by 0, the process terminates, and the obtained allocation is feasible, EF1 and SWM as desired. □

Acknowledgments

A. Dror and M.Feldman received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 866132), and the Israel Science Foundation (grant number 317/17). E. Segal-Halevi is supported by the Israel Science Foundation (grant number 712/20). We are grateful to Jonathan Turner, Jan Vondrak, Chandra Chekuri, Tony Huynh, Siddharth Barman and Arpita Biswas for their helpful comments.

References

Aleksandrov, M.; Aziz, H.; Gaspers, S.; and Walsh, T. 2015. Online fair division: analysing a food bank problem. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 2540–2546.

Amanatidis, G.; Birmpas, G.; Filos-Ratsikas, A.; Hollender, A.; and Voudouris, A. A. 2020. Maximum Nash Welfare and Other Stories About EFX. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 24–30. International Joint Conferences on Artificial Intelligence Organization.

Babaioff, M.; Ezra, T.; and Feige, U. 2020. Fair and Truthful Mechanisms for Dichotomous Valuations. *arXiv preprint arXiv:2002.10704* .

Barman, S.; Biswas, A.; Murthy, S. K. K.; and Narahari, Y. 2018. Groupwise Maximin Fair Allocation of Indivisible Goods. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 917–924. AAAI Press.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018. Greedy Algorithms for Maximizing Nash Social Welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 7–13.

Barrera, R.; Nyman, K.; Ruiz, A.; Su, F. E.; and Zhang, Y. X. 2015. Discrete envy-free division of necklaces and maps. ArXiv preprint 1510.02132.

Bei, X.; Igarashi, A.; Lu, X.; and Suksompong, W. 2019. The Price of Connectivity in Fair Division. ArXiv preprint 1908.05433.

Benabbou, N.; Chakraborty, M.; Igarashi, A.; and Zick, Y. 2020. Finding Fair and Efficient Allocations When Valuations Don't Add Up. In *Algorithmic Game Theory - 13th International Symposium, SAGT 2020, Proceedings*, volume 12283 of *Lecture Notes in Computer Science*, 32–46. Springer.

Bilò, V.; Caragiannis, I.; Flammini, M.; Igarashi, A.; Monaco, G.; Peters, D.; Vinci, C.; and Zwicker, W. S. 2018. Almost Envy-Free Allocations with Connected Bundles. In Blum, A., ed., *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 14:1–14:21. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-095-8. ISSN 1868-8969. doi:10.4230/LIPIcs.ITCS.2019.14. URL http://drops.dagstuhl.de/opus/volltexte/2018/10107.

Biswas, A.; and Barman, S. 2018. Fair Division Under Cardinality Constraints. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, 91–97. ijcai.org.

Biswas, A.; and Barman, S. 2019. Matroid Constrained Fair Allocation Problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9921–9922.

Bonin, J. E.; and Savitsky, T. J. 2016. An infinite family of excluded minors for strong base-orderability. *Linear Algebra and its Applications* 488: 396–429.

Bouveret, S.; Cechlárová, K.; Elkind, E.; Igarashi, A.; and Peters, D. 2017. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 135–141.

Bouveret, S.; and Lemaître, M. 2016. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems* 30(2): 259–290.

Brualdi, R. A.; and Scrimger, E. B. 1968. Exchange systems, matchings, and transversals. *Journal of Combinatorial Theory* 5(3): 244 – 257.

Budish, E. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119(6): 1061–1103.

Budish, E.; Cachon, G. P.; Kessler, J. B.; and Othman, A. 2017. Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research* 65(2): 314–336.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)* 7(3): 1–32.

Darmann, A.; and Schauer, J. 2015. Maximizing Nash product social welfare in allocating indivisible goods. *European Journal of Operational Research* 247(2): 548–559.

Dror, A.; Feldman, M.; and Segal-Halevi, E. 2020. On Fair Division under Heterogeneous Matroid Constraints. *arXiv preprint arXiv:2010.07280* .

Ferraioli, D.; Gourvès, L.; and Monnot, J. 2014. On regular and approximately fair allocations of indivisible goods. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 997–1004.

Garg, N.; Kavitha, T.; Kumar, A.; Mehlhorn, K.; and Mestre, J. 2010. Assigning papers to referees. *Algorithmica* 58(1): 119–136.

Gourvès, L.; and Monnot, J. 2019. On maximin share allocations in matroids. *Theoretical Computer Science* 754: 50–64. Preliminary version appeared in CIAC 2017.

Gourvès, L.; Monnot, J.; and Tlilane, L. 2013. A protocol for cutting matroids like cakes. In *International Conference on Web and Internet Economics*, 216–229. Springer.

Gourvès, L.; Monnot, J.; and Tlilane, L. 2014. Near Fairness in Matroids. In *ECAI 2014 - 21st European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, 393–398. IOS Press.

Halpern, D.; Procaccia, A. D.; Psomas, A.; and Shah, N. 2020. Fair Division with Binary Valuations: One Rule to Rule Them All. In *Web and Internet Economics - 16th International Conference, WINE 2020, Beijing, China, December 7-11, 2020, Proceedings*, volume 12495 of *Lecture Notes in Computer Science*, 370–383.

Klaus, B.; Manlove, D. F.; and Rossi, F. 2016. *Matching under preferences*. Cambridge University Press.

Lian, J. W.; Mattei, N.; Noble, R.; and Walsh, T. 2018. The Conference Paper Assignment Problem: Using Order Weighted Averages to Assign Indivisible Goods. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 1138–1145. AAAI Press.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM conference on Electronic commerce*, 125–131.

Long, C.; Wong, R. C.-W.; Peng, Y.; and Ye, L. 2013. On good and fair paper-reviewer assignment. In *2013 IEEE 13th International Conference on Data Mining*, 1145–1150. IEEE.

Mackin, E.; and Xia, L. 2016. Allocating indivisible items in categorized domains. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 359–365.

Nyman, K.; Su, F. E.; and Zerbib, S. 2020. Fair division with multiple pieces. *Discrete Applied Mathematics* 283: 115–122.

Okumura, Y. 2014. Priority matchings revisited. *Games and Economic Behavior* 88: 242–249.

Oxley, J. G. 2006. *Matroid theory*, volume 3. Oxford University Press, USA.

Roth, A. E.; Sönmez, T.; and Ünver, M. U. 2005. Pairwise kidney exchange. *Journal of Economic theory* 125(2): 151–188.

Sikdar, S.; Adali, S.; and Xia, L. 2017. Mechanism Design for Multi-Type Housing Markets. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 684–690.

Sikdar, S.; Adalı, S.; and Xia, L. 2019. Mechanism Design for Multi-Type Housing Markets with Acceptable Bundles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2165–2172.

Suksompong, W. 2019. Fairly allocating contiguous blocks of indivisible items. *Discrete Applied Mathematics* 260: 227–236.

Turner, J. 2015a. Faster Maximium Priority Matchings in Bipartite Graphs. *arXiv preprint arXiv:1512.09349* .

Turner, J. 2015b. Maximium Priority Matchings. *arXiv preprint arXiv:1512.08555* .