Envy-free Matchings in Bipartite Graphs and their Applications to Fair Division

Elad Aigner-Horev
Ariel University, Ariel 40700, Israel
horev.elad@gmail.com

Erel Segal-Halevi
Ariel University, Ariel 40700, Israel
erelsgl@gmail.com

Abstract

Envy-free matching is a relaxation of perfect matching. In a bipartite graph with parts X and Y, an envy-free matching is a matching of some vertices in X to some vertices in Y, such that each unmatched vertex in X is not adjacent to any matched vertex in Y (so the unmatched vertices do not "envy" the matched ones). The empty matching is always envy-free. This paper presents: (a) a sufficient condition for the existence of a nonempty envy-free matching; (b) a polynomial-time algorithm for finding a nonempty envy-free matching if-and-only-if it exists; (c) some applications of envy-free matching as a subroutine in fair division algorithms.

2012 ACM Subject Classification Mathematics of computing → Matchings and factors; Social and professional topics → Information system economics

Keywords and phrases Matching, Fair division, Bipartite graphs

Acknowledgements This paper benefited a lot from answers by Zur Luria and Yuval Filmus, and by comments from two anonymous referees to the WTAF 2019 conference.

1 Introduction

Consider a set of people, each of whom should be given a unique house. The people have binary preferences on the houses: each person either "accepts" or "rejects" each house. Ideally, we would like to give each person a house he/she accepts. In case this is impossible, we would like to find a partial solution, in which some people gain while the other people do not lose. Specifically, we would like some people to get a house they accept, while the other people do not accept any of the allocated houses. Thus, the "house-less" people do not envy the people that received a house.

In graph terms, we are given a bipartite graph $G = (X\dot{\cup}Y, E)$, in which X is the set of people, Y is the set of houses and E is the set of "accept" relations. We have to find a matching $M \subseteq E$, denoting an assignment of houses to people. Denote by X_M and Y_M the subsets of X and Y, respectively, matched by M. Ideally we would like M to be X-*saturating*, i.e., $X_M = X$. Since such a matching might not exist, we are interested in the following relaxation:

▶ **Definition 1.1.** *In a bipartite graph* $(X\dot{\cup}Y, E)$*, a matching* $M \subseteq E$ *between a subset* $X_M \subseteq X$ *and a subset* $Y_M \subseteq Y$ *is said to be* envy-free w.r.t. X *if every vertex* $x \in X \setminus X_M$ *is not adjacent to any vertex* $y \in Y_M$*.*

When X is clear from the context, we just say that M is *envy-free*. in an envy-free matching, an unmatched person $x \in X$ does not envy any matched person $x' \in X$, because x does not like any matched house $y' \in Y$ anyway.

Any X-saturating matching is envy-free, since all members of X are matched. But while the former may not exist, the latter always does: the empty matching is vacuously envy-free.

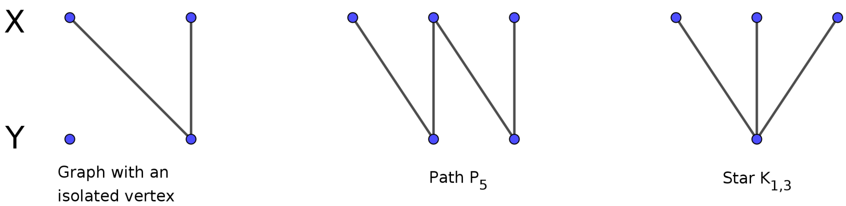

Figure 1 Some graphs in which the only envy-free matching (w.r.t X) is empty.

In some graphs, *only* the empty matching is envy-free; see Figure 1. This raises the question: when does a non-empty envy-free matching exist? Our first result provides a simple sufficient condition. To present it we need some definitions. for every subset $X' \subseteq X$, denote by $N_G(X')$ its neighborhood in Y, i.e.: $N_G(X') := \{y' \in Y : \exists x' \in X' \text{ such that } (x', y') \in E\}$.

▶ **Definition 1.2.** *A bipartite graph* $G := (X\dot{\cup}Y, E)$ *is called* weakly-Hall w.r.t. X *if* $|N_G(X)| \geq |X| \geq 1$.

When X is clear from the context, we just say that G is *weakly-Hall*. The weak-Hall condition is a relaxation of the condition in Hall's marriage theorem: the latter condition requires $|N_G(X')| \geq |X'|$ to hold for *any* subset $X' \subseteq X$, rather than just for X. While Hall's condition is sufficient (and necessary) for the existence of an X-saturating matching, the weak-Hall condition is sufficient for the existence of an X-envy-free matching:

▶ **Theorem 1.3.** *If* $G := (X\dot{\cup}Y, E)$ *is weakly-Hall, then* G *contains a nonempty envy-free matching, and such matching can be found in polynomial time.*

The existential part of Theorem 1.3 was proved by [13]. We present a constructive proof using a polynomial-time algorithm in **Section 2**.

Our second result provides a decision algorithm for general bipartite grpahs:

▶ **Theorem 1.4.** *There is a polynomial-time algorithm that, given any bipartite graph* $G := (X\dot{\cup}Y, E)$*, decides whether* G *contains a nonempty envy-free matching, and if so, finds one such matching.*

The algorithm is presented in **Section 3**. Note that the algorithm uses, as a subroutine, the algorithm of Theorem 1.3, so both theorems are needed.

Envy-free matching is not entirely a new invention: we have seen similar ideas (particularly, similar to Theorem 1.3) in previous papers. However, they were "hidden" inside more complex algorithms. Our goal in presenting envy-free matching as a stand-alone concept is twofold.

First, we would like to make the concept more visible and easily usable as a subroutine in future fair division algorithms. As an illustration, some old and new algorithms for fair division using envy-free matching as a subroutine are presented in **Section 4**.

Second, envy-free matching may be an interesting graph-theoretic concept in itself, as a relaxation of perfect matching. It invokes various algorithmic problems that, as far as we know, have not been studied before. The most basic one is the *decision* problem — deciding whether a nonempty envy-free matching exists. This problem is solved efficiently by our Theorem 1.4. Some more advanced algorithmic problems, that we plan to study in future work, are presented in **Section 5**.

1.1 A note on terminology

The term *envy-free matching* is used, in a somewhat more specific sense, in the context of markets, both with and without money.

In a market with money, there are several buyers and several goods, and each good may have a price. Given a price-vector, an “envy-free matching” is an allocation of bundles to agents in which each agent weakly prefers his bundle over all other bundles (given their respective prices). This is a relaxation of a *Walrasian equilibrium.* A Walrasian equilibrium is an envy-free matching in which every item with a positive price is allocated to some agent. In a Walrasian equilibrium, the seller’s revenue might be low. This motivates its relaxation to envy-free matching, in which the seller may set reserve-prices (and leave some items with positive price unallocated) in order to increase his expected revenue. See, for example, [10, 1].

In a market without money, there are several people who should be assigned to positions. For example, several doctors have to be matched for residency in hospitals. Each doctor has a preference-relation on hospitals (ranking the hospitals from best to worst), and each hospital has a preference relation on doctors. Each doctor can work in at most one hospital, and each hospital can employ at most a fixed number of doctors (called the capacity of the hospital). A matching has *justified envy* if there is a doctor d and a hospital h, such that d prefers h over his current employer, and h prefers d over one of its current employees. An “envy-free matching” is a matching with no justified envy. This is a relaxation of a *stable matching.* A stable matching is an envy-free matching which is also non-wasteful — there is no doctor d and a hospital h, such that d prefers h over his current employer and h has some vacant positions [19]. When the hospitals have, in addition to upper quotas (capacities), also *lower quotas*, a stable matching might not exist. This motivates its relaxation to envy-free matching, which always exists and can be found efficiently [20].

In contrast to these works, our envy-free matching is defined for *unweighted* bipartite graphs.

2 Sufficient Condition

This section proves Theorem 1.3. The proof uses Hall-violators. A *Hall violator* is a subset $X_H \subseteq X$ for which $|N_G(X_H)| < |X_H|$. The term comes from Hall’s famous *marriage theorem*, which says that G admits an X-saturating matching if-and-only-if X contains no Hall violators. Hall’s theorem can be proved by an algorithm that, given a maximum-cardinality matching $M \subseteq E$ that does *not* saturate X, finds a Hall-violator in X. Implementing such algorithm is a standard exercise in graph theory. For completeness, we present a (somewhat more general) algorithm as Algorithm 5 in Appendix A. The run-time complexity of that algorithm is linear in the size of the sub-graph spanned by its output, that is, linear in the size of $G[X_H, N_G(X_H)]$. Using Algorithm 5 as a subroutine, Algorithm 1 finds an envy-free matching in a weakly-Hall graph.

Since a Hall violator is non-empty, the graph shrinks with every iteration so the algorithm must eventually terminate (at step 2). To justify the returned value at that step, observe that a weakly-Hall graph always contains a non-empty matching, for example, a single edge between a vertex in X to its neighbor in Y. Hence, the maximum-cardinality matching M is non-empty. Since it is X-saturating, it is envy-free.

To justify step 3, we have to consider what happens to the weakly-Hall graph after a Hall-violator and its neighbors are removed. We prove that (a) the smaller graph is still weakly-Hall, (b) an envy-free matching in the smaller graph is envy-free in the original graph too, and (c) the remainder of a maximum matching in the original graph is a maximum

Algorithm 1 Find a nonempty envy-free matching in a weakly-Hall graph

Input: A bipartite graph $G := (X\dot{\cup}Y, E)$ with $1 \leq |X| \leq |N_G(X)|$.
Output: A nonempty envy-free matching w.r.t. X.
1: Initialize M to a maximum cardinality matching in G.
2: If M saturates X, then **return** M.
3: Otherwise, use Algorithm 5 to find a Hall-violator $X_H \subseteq X$; denote $Y_H := N_G(X_H)$:
- Remove X_H and their neighbors from G, i.e. set $X := X \setminus X_H$ and $Y := Y \setminus Y_H$;
- Remove the edges between X_H and Y_H from M and go back to step 2.

matching in the smaller graph. This is done in the following lemma.

▶ **Lemma 2.1.** *Let $G := (X\dot{\cup}Y, E)$ be weakly-Hall and let $X_H \subseteq X$ be a Hall-violator. Let $Y_H = N_G(X_H)$ and $X_C := X \setminus X_H$ and $Y_C := Y \setminus Y_H$ and $G_C := G[X_C, Y_C]$ = the graph induced by X_C and Y_C. Then:*
(a) G_C is weakly-Hall (w.r.t. X_C);
(b) Any envy-free matching in G_C is an envy-free matching in G (w.r.t X);
(c) If M is a maximum matching in G, then $M[X_C, Y_C]$ is a maximum matching in G_C.

Proof. (b) The matching is envy-free in G_C, so vertices of X_C do not envy other vertices in X_C. By construction, there are no edges between vertices in X_H and vertices in Y_C, so vertices of X_H do not envy the vertices in X_C. By construction, vertices outside X_C are unmatched, so no other vertex envies them.

(a) We first prove that $|X_C| \geq 1$. Indeed, since G is weakly-Hall, $|N_G(X)| \geq |X|$, so X itself is not a Hall-violator. Hence the Hall-violator X_H must be a strict subset of X.

It remains to prove that $|N_{G_C}(X_C)| \geq |X_C|$. $N_G(X)$ can be partitioned into two parts: $N_G(X) \cap Y_C$ and $N_G(X) \cap Y_H$. The first part equals $N_G(X_C) \cap Y_C$, since there are no edges between X_H and Y_C. Hence:

$$\begin{aligned}
|N_G(X_C) \cap Y_C| + |N_G(X) \cap Y_H| &= |N_G(X)| \geq |X| && \text{since } G \text{ is weakly-Hall} \\
&= |X_H| + |X_C| \\
&> |N_G(X_H)| + |X_C| && \text{since } X_H \text{ is a Hall-violator} \\
&= |N_G(X) \cap Y_H| + |X_C|.
\end{aligned}$$

This implies $|N_G(X_C) \cap Y_C| \geq |X_C|$, which is equivalent to $|N_{G_C}(X_C)| \geq |X_C|$.

(c) Suppose that $M_C := M[X_C, Y_C]$ is not a maximum matching in G_C. By Berge's lemma, there is an M_C-augmenting path in G_C. The same path is M-augmenting in G, so M is not a maximum matching in G. ■

We now estimate the run-time complexity of Algorithm 1. The maximum matching required in step 1 can be found, for example, by the Hopcroft-Karp algorithm in time $O(\sqrt{|V|}|E|)$. Algorithm 5 runs in time linear to the size of the removed edges and vertices. Hence, the run-time of the loop in steps 2 and 3 is in $O(|E|)$.

3 Decision Algorithm

This section proves Theorem 1.4 by Algorithm 2. Below we justify each step in turn.

Step 1 removes *isolated vertices* — vertices with no adjacent edges. Such vertices cannot participate in any matching, so removing them does not change the set of envy-free matchings. If X is now empty, then obviously no nonempty matching exists.

Algorithm 2 Find a nonempty envy-free matching in a bipartite graph iff it exists

Input: A bipartite graph $G := (X\dot{\cup}Y, E)$.

Output: A nonempty envy-free matching w.r.t. X, if one exists. Otherwise, **None**.

1: Remove isolated vertices from X and Y. If $X = \emptyset$, return **None**.

2: If $|Y| \geq |X|$, then use Algorithm 1 to find a nonempty envy-free matching in G, and return it.

3: If $|Y| < |X|$, then find either a Y-saturating matching or a Hall violator in Y.

4: If a Hall violator $Y_H \subseteq Y$ is found, then define $X_H := N_G(Y_H)$ and $G_H := G[X_H, Y_H]$, use Algorithm 1 to find a nonempty envy-free matching in G_H, and return it.

5: If a Y-saturating matching is found, denote the subset of X matched to Y by X_M. Denote $Y_M := Y \setminus N_G(X \setminus X_M)$. Recurse on the graph $G_M := G[X_M, Y_M]$.

Once there are no isolated vertices, we have $N_G(X) = Y$. Hence, $|Y| \geq |X|$ combined with $X \neq \emptyset$ implies that G is a weakly-Hall graph. Hence, in step 2, Algorithm 1 finds and returns a nonempty envy-free matching.

Step 3 requires more elaboration. First, we find a maximum-cardinality matching. If its cardinality is $|Y|$, then it is Y-saturating and we proceed to step 5 below. Otherwise, Algorithm 5 (used in the opposite direction to which it was used in Algorithm 1) returns a Hall violator in Y — a subset $Y_H \subseteq Y$ such that $|N_G(Y_H)| < |Y_H|$, and we proceed to step 4.

The correctness of the next steps is proved by the following lemmas.

The next lemma justifies Step 4.

▶ **Lemma 3.1.** *Let $G = (X\dot{\cup}Y, E)$ by a bipartite graph with no isolated vertices, and let $Y_H \subseteq Y$ be a Hall-violator. Let $X_H := N_G(Y_H)$ and $G_H := G[X_H, Y_H]$ = the graph induced by X_H and Y_H. Then:*

(a) G_H is weakly-Hall (w.r.t. X_H);

(b) Any envy-free matching in G_H is an envy-free matching in G (w.r.t. X).

Proof. (a) Since Y_H is nonempty, and G has no isolated vertices, X_H is nonempty too. Moreover, $N_G(X_H) = N_G(N_G(Y_H)) \supseteq Y_H$ so $|N_G(X_H)| \geq |Y_H| > N_G(Y_H) = |X_H|$. Hence G_H is weakly-Hall.

(b) The matching is envy-free in G_H, so vertices of X_H do not envy other vertices in X_H. By construction, there are no edges between vertices in $X \setminus X_H$ and vertices in Y_H, so vertices of $X \setminus X_H$ do not envy vertices of X_H. By construction, vertices outside X_H are unmatched, so no other vertex envies them. ■

The next couple of lemmas justify Step 5.

▶ **Lemma 3.2.** *Let $G = (X\dot{\cup}Y, E)$ by a bipartite graph admitting a Y-saturating matching M. Let N an envy-free matching in G (w.r.t. X). Then, every vertex of X matched in N is matched in M too.*

Proof. Let $x_0 \in X$ be a vertex matched in N to some vertex $y_1 \in Y$. Suppose by contradiction that x_0 is not matched in M. Since M is Y-saturating, y_1 is matched in M to some other vertex $x_1 \in X$, $x_1 \neq x_0$. In particular, x_1 is adjacent to y_1 which is matched by N to x_0. To ensure that x_1 is not envious, it must be matched in N too, and it must be matched to a different vertex, say $y_2 \in Y$, $y_2 \neq y_1$. Since M is Y-saturating, y_2 must be matched in M to some other vertex $x_2 \in X$ (see Figure 2). This argument goes on ad infinitum, and no vertices can be repeated since both M and N are matchings. However, the graph is finite — a contradiction. ■

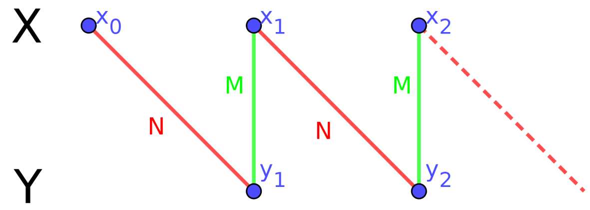

Figure 2 Illustration of the proof of Lemma 3.2.

▶ **Lemma 3.3.** *Let $G = (X\dot{\cup}Y, E)$ by a bipartite graph admitting a Y-saturating matching between Y and a subset $X_M \subseteq X$. Let $G_M := G[X_M, Y \setminus N_G(X \setminus X_M)]$. Then, the set of envy-free matchings in G_M is equal to the set of envy-free matchings in G.*

Proof. Let N be an envy-free matching in G. The vertices in $X \setminus X_M$ are unmatched in M, so by Lemma 3.2, they are unmatched in N too. To ensure that the vertices in $X \setminus X_M$ do not envy, the vertices in $N_G(X \setminus X_M)$ must also be unmatched in N. Hence, N is a matching in G_M too (it is obviously envy-free in G_M, since G_M is a subset of G).

Conversely, let N be a matching that is envy-free in G_M. When N is treated as a matching in G, the vertices of X_M still do not envy. By construction, there are no edges between vertices of $X \setminus X_M$ and vertices of $Y \setminus N_G(X \setminus X_M)$, so vertices of $X \setminus X_M$ do not envy. Hence, N is envy-free in G too. ∎

Lemma 3.3 implies that, whenever there is a Y-saturating matching that leaves some vertices in X unmatched, we can remove these unmatched vertices from X and remove their neighbors from Y, and get a smaller graph with the same set of envy-free matchings. This justifies the recursion in Step 5 and completes the proof of correctness of Algorithm 2.

4 Related Work and Applications

4.1 Envy-free matching in fair cake-cutting

As far as we know, the earliest concept similar to envy-free matching was presented in a lemma of [11][page 31]. He presents the lemma in matrix notation. In graph terminology, his lemma says that an envy-free matching exists whenever $|X| = |Y|$ and there is a vertex $x \in X$ for whom $N_G(\{x\}) = |Y|$. This is a special case of our Theorem 1.3. Kuhn uses this lemma to extend a method of [18] for *fair cake-cutting.*

In a fair cake-cutting problem, there is a heterogeneous and continuous resource C ("cake"), usually thought of as the real interval $[0, 1]$. There are n agents and each agent i has a nonatomic value-measure V_i on C. A *proportional division* of C is a partition into n pairwise-disjoint pieces, $C = X_1 \sqcup \cdots \sqcup X_n$, such that $\forall i \in [n] : V_i(X_i) \geq V_i(C)/n$. There are various algorithms for proportional cake-cutting. One of them is now called the *lone divider* method. Steinhaus presented it for 3 agents. [11] extended it to any number of agents. The cases $n = 3$ and $n = 4$ are described in detail by [4][pages 31-35], and the general case is described in detail by [16][pages 83-87]. To demonstrate the usefulness of envy-free matching, we use it to present the lone divider method in a simpler way; see Algorithm 3.

To see the correctness of the algorithm, observe that:

- In step 2, Alice is adjacent to all n pieces, so $|N_G(\text{Agents})| = n \geq |\text{Agents}|$, so the precondition for Algorithm 1 is met and it a nonempty envy-free matching can be found.

Algorithm 3 Cut a cake fairly using the Lone Divider method.

Input: a cake C and $n \geq 1$ agents with nonatomic measures $(V_i)_{i=1}^n$ on C, normalized such that $\forall i \in [n] : V_i(C) \geq n$.

Output: a partition $C = X_1 \sqcup \cdots \sqcup X_n$ such that $\forall i \in [n] : V_i(X_i) \geq 1$.

1: Pick one agent arbitrarily, say Alice, and ask her to cut C into n disjoint pieces $C_1, \ldots, C_n$ with an equal value for her.

2: Create a bipartite graph G with the agents in one side, the pieces in the other side, and agent i is adjacent to piece C_j iff $V_i(C_j) \geq 1$.

3: Find a nonempty envy-free matching in G. Give each matched piece to its agent.

4: Let $l \geq 1$ be the number of matched agents. If $l = n$, return. Otherwise, recurse with the remaining cake and the remaining $n - l$ agents.

- In step 3, each matched agent receives a piece with a value of at least 1, and so the division is proportional for that agent.
- In step 4, since the matching is envy-free, each remaining agent values each piece given away as less than 1, so for every remaining agent i, V_i of the remaining cake is more than $n - l$. Hence the precondition for Algorithm 3 is satisfied and it can run recursively.

One may ask why is the Lone Divider method needed when there are many other algorithms for proportional cake-cutting, such as [18] and [8]? The reason is that Lone Divider can be used as a subroutine in more advanced fair division algorithms. As a recent example, [6] presented an algorithm that is not only fair but also *aristotelian* — it guarantees that agents with an identical value-measure always get an equal value. He shows that the classic cake-cutting algorithms are not aristotelian. His algorithm uses Lone Divider as a subroutine,

Moreover, in [17] the second author has found envy-free matching useful in solving a variant of the fair cake-cutting problem called *archipelago division*. In that variant, C is made of m disconnected islands, and we need to give each agent a subset of C that overlaps at most $k < m$ islands, so that an agent does not have to jump between too many different islands to get from one side of his plot to the other side. The case $k = 1$ is easy to solve, but the case $k = 2$ is more challenging. It can be solved optimally using envy-free matching.

4.2 Envy-free matching in fair allocation of discrete objects

In [15][subsection 3.1], we found another concept similar to envy-free matching, hiding inside proofs of lemmas related to fair allocation of discrete (indivisible) objects.[1] This problem involves a finite set O of objects. There are n agents. Each agent i has a value measure (i.e, an additive function) $V_i : 2^O \to \mathbb{R}_+$.

Since the objects are discrete, a proportional allocation might not exist. Hence, the following relaxation is used. For every agent $i \in [n]$ and integers $1 \leq l \leq d$, define the *l-out-of-d maximin-share of i from O* as:

$$M_i\left(\frac{l}{d}O\right) := \max_{\mathbf{Y} \in \textsc{Partitions}(O,d)} \quad \min_{Z \in \textsc{Union}(\mathbf{Y},l)} \quad V_i(Z)$$

where the maximum is over all partitions of O into d subsets, and the minimum is over all unions of l subsets from the partition. $M_i\left(\frac{l}{d}O\right)$ is the largest value that agent i can get by

[1] This paper has a journal version [12] where these lemmas do not appear.

dividing O into d piles and getting the worst l piles. Obviously $M_i\left(\frac{l}{d}O\right) \leq \frac{l}{d}V_i(O)$, with equality holding iff O can be partitioned into d subsets with the same value. Thus, $M_i\left(\frac{l}{d}O\right)$ can be thought of informally as $\frac{l}{d}V_i(O)$ "rounded down to the nearest object".

The maximin share (with $l = 1$) was introduced by [5]. The generalization to any $l \geq 1$ was introduced by [2].

[5] asked whether there always exists a partition of O in which the value of each agent i is at least $M_i\left(\frac{1}{n}O\right)$. For $n = 2$, the answer is 'Yes' and such an allocation is easily found:

- Alice partitions O into two piles that she values as at least $M_A\left(\frac{1}{2}O\right)$ (in other words, she finds a partition $\mathbf{Y}$ attaining the maximum in the definition of $M_A\left(\frac{1}{2}O\right)$). [2]
- Bob chooses the pile he prefers and Alice receives the other pile.

For $n \geq 3$, [15] and [12] prove that such allocation might not exist. They present a *multiplicative approximation* algorithm, by which each agent i receives a bundle whose value is at least $\gamma \cdot M_i\left(\frac{1}{n}O\right)$, for some fraction $\gamma \in (0, 1)$. Their approximation fraction γ equals $3/4$ for $n \in \{3, 4\}$, and approaches $2/3$ when $n \to \infty$.

They then consider an *additive approximation* — a partition in which the value of each agent i is at least $M_i\left(\frac{1}{n+1}O\right)$. They say that

> "We have designed an algorithm that achieves this guarantee for the case of three players (it is already nontrivial). Proving or disproving the existence of such allocations for a general number of players remains an open problem."

They do not present the algorithm for $n = 3$.[3] Using envy-free matching, we could design such an algorithm. It is presented as Algorithm 4.

Algorithm 4 Find a 1-out-of-4 MMS allocation among 3 agents.

Input: a set O of goods and $n = 3$ agents with measures $(V_i)_{i=1}^n$ on O.
Output: a partition $X_1 \sqcup X_2 \sqcup X_3$ such that $\forall i \in [n] : V_i(X_i) \geq M_i\left(\frac{1}{4}O\right)$.

1: Pick one agent arbitrarily, say Alice, and ask her to divide O into 3 disjoint piles $(O_i)_{i=1}^3$ such that $\forall i \in [3] : V_A(O_i) \geq M_A\left(\frac{1}{4}O\right)$.
2: Create a bipartite graph G with the agents in one side, the piles in another side, and agent i is adjacent to pile O_j iff $V_i(O_j) \geq M_i\left(\frac{1}{4}O\right)$.
3: Find a nonempty envy-free matching in G. Give each matched pile to its agent.
4: Let $l \geq 1$ be the number of matched agents. If $l = 3$, return.
If $l = 2$, give the remaining goods to the single remaining agent.
If $l = 1$, divide the remainder among the remaining agents by cut-and-choose.

The algorithm is very similar to the Lone Divider method for cake-cutting. In step 2, Alice is adjacent to all piles, so the precondition for Algorithm 1 holds and a nonempty envy-free matching is found in step 3.

Step 4 requires more explanation. Since the matching is envy-free, each remaining agent values each allocated pile as less than $M_i\left(\frac{1}{4}O\right)$. When $l = 2$, there is one remaining agent and he values the remaining goods as at least $V_i(G) - 2M_i\left(\frac{1}{4}O\right) \geq 4M_i\left(\frac{1}{4}O\right) - 2M_i\left(\frac{1}{4}O\right) > M_i\left(\frac{1}{4}O\right)$.

[2] To find this maximizing partition, Alice may have to solve the NP-hard problem Partition. However, this is not the problem of the division manager. In fact, if Alice is computationally bounded and provides a sub-optimal partition, this makes life even easier for the division manager — he can give Alice a smaller value and she cannot complain.

[3] Perhaps they wanted to write it in the margin but the margin was too narrow ☺

For $l = 1$, we need to show that for each remaining agent, the 1-out-of-2 maximin share of the remaining goods is at least $M_i\left(\frac{1}{4}O\right)$. So we need to show that each agent can partition the remaining goods into two piles with value at least $M_i\left(\frac{1}{4}O\right)$. Let i be a remaining agent. Let $H_1 \sqcup H_2 \sqcup H_3 \sqcup H_4$ be his 1-out-of-4 maximin partition of G. For $j \in [4]$, let H_j^A be the subset of H_j that Alice took in Step 3, and $H_j^B := H_j \setminus H_j^A$ the remaining subset. Now $V_i(H_1^B \cup H_2^B) \geq V_i(H_1) + V_i(H_2) - M_i\left(\frac{1}{4}O\right) \geq 2M_i\left(\frac{1}{4}O\right) - M_i\left(\frac{1}{4}O\right) = M_i\left(\frac{1}{4}O\right)$. Similarly, $V_i(H_3^B \cup H_4^B) \geq M_i\left(\frac{1}{4}O\right)$. Hence, the 2-partition $(H_1^B \cup H_2^B, H_3^B \cup H_4^B)$ shows that the 1-out-of-2 MMS of i is at least $M_i\left(\frac{1}{4}O\right)$. The same is true for both agents. Hence, divide-and-choose on the remaining goods guarantees each agent i at least $M_i\left(\frac{1}{4}O\right)$. [4] ◀

Algorithm 4 can be modified to give various other fairness guarantees. For example, it can guarantee to each agent i at least his 2-out-of-7 maximin share, i.e., $V_i(X_i) \geq M_i\left(\frac{2}{7}O\right)$. In some cases, this guarantee may be better than the 1-out-of-4 MMS. For example, if G contains 7 items and agent i values all of them as 1, then $M_i\left(\frac{2}{7}O\right) = 2$ while $M_i\left(\frac{1}{4}O\right) = 1$. Alternatively, Algorithm 4 can guarantee a multiplicative approximation, giving each agent i a value of at least $\frac{3}{4}M_i\left(\frac{1}{3}O\right)$. It can even make different guarantees to different agents. The only required change is that in step 2, in the bipartite graph, each agent i should be adjacent to pile O_j iff $V_i(O_j)$ is above the fairness threshold chosen by agent i (e.g. $V_i(O_j) \geq M_i\left(\frac{2}{7}O\right)$ or $V_i(O_j) \geq \frac{3}{4}M_i\left(\frac{1}{3}O\right)$). For each fairness criterion, we have to prove that, in step 4, if a subset whose value is below the threshold was taken, then the remaining goods can be partitioned into two subsets both of which are above the threshold. The proof for each criterion is similar to the one for $M_i\left(\frac{1}{4}O\right)$ and we omit it.[5]

While there are now algorithms that attain better multiplicative approximation factors (e.g. [3, 9]), it is nice to have a simple algorithm using envy-free matching that can handle multiplicative and additive approximations simultaneously.

5 Extensions and Future Work

We are currently working on improving and extending our results in various directions.

1. Instead of requiring each vertex in X ("agent") to match at most one vertex in Y ("house"), we can consider allocation problems in which each vertex in X can be connected to any number of vertices in Y (but each vertex in Y can still be connected to at most one vertex in X). A vertex x_1 envies another vertex x_2 iff there are more vertices matched to x_2 that are acceptable to x_1, then vertices matched to x_1.
2. This paper focuses on finding a *nonempty* envy-free matching. For the fair division algorithms presented in Section 4, nonemptiness is sufficient since it guarantees that the algorithm makes a progress in each iteration. However, for other applications it may be useful to find a *maximum-cardinality* envy-free matching. We are now working on an algorithm for that problem.
3. As indicated by the works referenced in subsection 1.1, it may be interesting to extend the envy-free matching concept from unweighted graphs (where each edge represents

---

[4] To show the difficulty in extending this algorithm to four or more agents, here is an example scenario. There are 4 agents and 10 goods that some agent, George, values as 5,5,5,5,5,1,1,1,1,1, so $M_G\left(\frac{1}{5}O\right) = 6$. In the first round, Alice makes four piles, putting all five goods that George values at 1 in one pile. George is not adjacent to this pile since it is worth less than 6. Suppose no other agent is adjacent to this pile, so Alice takes it. In the second round, it's George's turn to divide, but he cannot make three piles each of which is worth at least 6.

[5] The main challenge in extending this algorithm to 4 or more agents is in step 4 when $l = 1$.

an "acceptability" relation), to weighted graphs (where the weights represent ordinal or cardinal preferences).

4. From a graph-theoretic perspective, it may be interesting to generalize envy-free matchings from bipartite to general graphs.
5. It may be interesting to calculate the probability that a non-empty envy-free matching exists in a random graph. This is related to the problem of calculating the probability that an envy-free allocation exists, which has recently been studied by e.g. [7] and [14].

A Finding a Hall violator

Algorithm 1 requires a subroutine that, given a maximum-cardinality matching that is not X-saturating, finds a Hall-violator in X. Below we present a more general algorithm, that can receive as input any matching M. When M is maximum-cardinality it always yields a Hall-violator; otherwise it may return either a Hall-violator or an M-augmenting path.

Algorithm 5 Find either a Hall-violator or an augmenting path.

Input: a graph $G = (X\dot{\cup}Y, E)$; a matching $M \subseteq E$; an unmatched vertex $x_0 \in X \setminus V(M)$.
Output: one of two options:

- a. An M-augmenting path starting at x_0, or —
- b. A Hall-violator containing x_0 — a subset $X_H \subseteq X$ for which $|N_G(X_H)| < |X_H|$.

1: Initialize $k := 0$, $X_k := \{x_0\}$, $Y_k := \{\}$.
2: If $N_G(X_k) \subseteq Y_k$ then **return** X_k as a Hall-violator.
3: Otherwise, let y_{k+1} be a vertex in $N_G(X_k) \setminus Y_k$.
4: If y_{k+1} is matched by M, then let $x_{k+1} \in X$ be its partner in M:
- Let $X_{k+1} = X_k \cup \{x_{k+1}\}$ and $Y_{k+1} = Y_k \cup \{y_{k+1}\}$ and $k = k + 1$.
- Go back to step 2.

5: Otherwise (y_{k+1} is unmatched by M), construct an M-augmenting path from y_{k+1} to x_0 (see below); **return** one such path.

To justify Algorithm 5, we first prove that the following are true whenever the algorithm is before step 2:

- (a) $X_k = \{x_0, ..., x_k\}$ where the x_i are distinct vertices of X;
- (b) $Y_k = \{y_1, ..., y_k\}$ where the y_i are distinct vertices of Y;
- (c) For all $i \geq 1$, y_i is matched by M to x_i;
- (d) For all $i \geq 1$, y_i is connected to some x_j, for $j < i$, by an edge not in M.

The proof is by induction. After the initialization step, all four claims are true (some vacuously). We have to prove that, in step 4, the vertex x_{k+1} is a new vertex that is not in X_k. Indeed, x_0 is unmatched by M, and every $x_i \in X_k$ is matched to $y_i \in Y_k$ (by the induction assumption), and $y_{k+1} \notin Y_k$ by its definition. Therefore, the partner of y_{k+1} in M must be a vertex from $x \setminus X_k$. Since $y_{k+1} \in N_G(X_k)$ by its definition, it is connected to some x_i (for $i < k + 1$), and the connecting edge cannot be in M.

By (a) and (b), the set X_k is always larger than Y_k by one vertex. Hence, in step 2, if $N_G(X_k) \subseteq Y_k$, then $|N_G(X_k)| \leq |Y_k| = |X_k| - 1 < |X_k|$, so X_k is indeed a Hall violator.

It remains to justify step 5. The vertex y_{k+1}, is connected to some x_i (for $i < k + 1$) by an edge not in M. By (c), this x_i is connected to y_i by an edge in M. By (d), y_i is connected to some x_j (for $j < i$) by an edge not in M, and so on. Following these connections must eventually lead to x_0, which is unmatched. This process yields an M-augmenting path. ◀

References

1 Saeed Alaei, Kamal Jain, and Azarakhsh Malekian. Competitive Equilibria in Two Sided Matching Markets with Non-transferable Utilities, December 2012. arXiv preprint 1006.4696. URL: `http://arxiv.org/abs/1006.4696`, `arXiv:1006.4696`.

2 Moshe Babaioff, Noam Nisan, and Inbal Talgam-Cohen. Competitive Equilibria with Indivisible Goods and Generic Budgets, March 2017. arXiv preprint 1703.08150. URL: `http://arxiv.org/abs/1703.08150`, `arXiv:1703.08150`.

3 Siddharth Barman and Sanath Kumar Krishna Murthy. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 647–664. ACM, 2017.

4 Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake Cutting to Dispute Resolution*. Cambridge University Press, Cambridge UK, February 1996.

5 Eric Budish. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, 119(6):1061–1103, December 2011. URL: `http://dx.doi.org/10.1086/664613`, `doi:10.1086/664613`.

6 Guillaume Chèze. Don't cry to be the first! symmetric fair division algorithms exist, 2018. arXiv preprint 1804.03833.

7 John P. Dickerson, Jonathan Goldman, Jeremy Karp, Ariel D. Procaccia, and Tuomas Sandholm. The Computational Rise and Fall of Fairness. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1405–1411. AAAI Press, 2014. URL: `http://portal.acm.org/citation.cfm?id=2894091`.

8 Shimon Even and Azaria Paz. A Note on Cake Cutting. *Discrete Applied Mathematics*, 7(3):285–296, March 1984. URL: `http://dx.doi.org/10.1016/0166-218x(84)90005-2`, `doi:10.1016/0166-218x(84)90005-2`.

9 Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 539–556. ACM, 2018. arXiv preprint 1704.00222.

10 Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On Profit-maximizing Envy-free Pricing. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1164–1173, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics. URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.9450&rep=rep1&type=pdf`.

11 Harold W. Kuhn. *On games of fair division*, pages 29–37. Princeton University Press, 1967.

12 David Kurokawa, Ariel D Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM (JACM)*, 65(2):8, 2018.

13 Zur Luria. Envy-free matching. Mathematics Stack Exchange, December 2013. URL: https://math.stackexchange.com/q/621306/29780 (version: 2013-12-29). URL: `https://math.stackexchange.com/q/621306/29780`.

14 Pasin Manurangsi and Warut Suksompong. When do envy-free allocations exist? In *Proceedings of AAAI'19*, 2018. arXiv preprint 1811.01630.

15 Ariel D. Procaccia and Junxing Wang. Fair Enough: Guaranteeing Approximate Maximin Shares. In *Proc. EC-14*, pages 675–692, New York, NY, USA, 2014. ACM. URL: `http://www.cs.cmu.edu/~{}arielpro/papers/mms.pdf`, `doi:10.1145/2600057.2602835`.

16 Jack M. Robertson and William A. Webb. *Cake-Cutting Algorithms: Be Fair if You Can*. A K Peters/CRC Press, first edition, July 1998.

17 Erel Segal-Halevi. Fair division of an archipelago, 2018. arXiv preprint 1812.08150.

18 Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, jan 1948. URL: `http://www.jstor.org/stable/1914289`.

19 Qingyun Wu and Alvin E Roth. The lattice of envy-free matchings. *Games and Economic Behavior*, 109:201–211, 2018.

20 Yu Yokoi. Envy-free matchings with lower quotas. *arXiv preprint 1704.04888*, 2017.