



Envy-free matchings in bipartite graphs and their applications to fair division

Elad Aigner-Horev, Erel Segal-Halevi *

Ariel University, Kiriath Hamada 3, Ariel 40700, Israel

ARTICLE INFO

Article history:

Received 22 June 2020

Received in revised form 1 July 2021

Accepted 19 November 2021

Available online 27 November 2021

Keywords:

Fair division

Cake cutting

Maximin share

Bipartite graphs

Perfect matching

Maximum matching

ABSTRACT

A matching in a bipartite graph with parts X and Y is called *envy-free*, if no unmatched vertex in X is adjacent to a matched vertex in Y . Every perfect matching is envy-free, but envy-free matchings exist even when perfect matchings do not. We prove that every bipartite graph has a unique partition such that all envy-free matchings are contained in one of the partition sets. Using this structural theorem, we provide a polynomial-time algorithm for finding an envy-free matching of maximum cardinality. For edge-weighted bipartite graphs, we provide a polynomial-time algorithm for finding a maximum-cardinality envy-free matching of minimum total weight. We show how envy-free matchings can be used in various fair division problems with either continuous resources (“cakes”) or discrete ones. In particular, we propose a symmetric algorithm for proportional cake-cutting, an algorithm for 1-out-of- $(2n-2)$ maximin-share allocation of discrete goods, and an algorithm for 1-out-of- $\lfloor 2n/3 \rfloor$ maximin-share allocation of discrete bads among n agents.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Let $G := (X \cup Y, E)$ be a bipartite graph. A matching $M \subseteq E$ is called *perfect* if every vertex of $X \cup Y$ is adjacent to exactly one edge of M ; it is called *X-saturating* if every vertex of X is adjacent to exactly one edge of M . This paper studies the following relaxation of *X-saturating* matching (where X_M and Y_M denote the vertices of X and Y , respectively, that are incident to edges of M).

Definition 1.1. Let $G := (X \cup Y, E)$ be a bipartite graph. A matching $M \subseteq E$ is said to be *envy-free w.r.t. X* if no vertex in $X \setminus X_M$ is adjacent to any vertex in Y_M .

One may view X as a set of people and Y as a set of houses, where a person in X is adjacent to all houses in Y which he or she likes. A matching $M \subseteq E$ denotes an assignment of houses to people who like them. Throughout the paper, all envy-free matchings are taken w.r.t. X . In such a matching, an unmatched person $x \in X \setminus X_M$ does not envy any matched person $x' \in X_M$, because x does not like any matched house $y' \in Y_M$ anyway.

If a matching M is *X-saturating*, then $X_M = X$, and M is clearly envy-free. A graph admitting an *X-saturating* matching is called *X-saturated*; see Fig. 1(a). Many graphs are not *X-saturated*, but still admit a non-empty envy-free matching; see Fig. 1(b).

* Corresponding author.

E-mail address: erelsgl@gmail.com (E. Segal-Halevi).

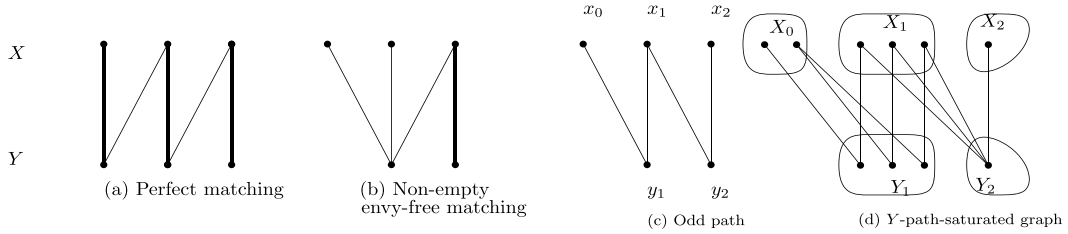


Fig. 1. (a) An X -saturated graph. (b) A graph that is not X -saturated, but admits a non-empty envy-free matching, denoted in bold. (c), (d) Graphs in which the only envy-free matching is empty.

In contrast, in some graphs the only envy-free matching is the empty matching \emptyset (which is vacuously envy-free). A natural example is an *odd path*—a path with $2k + 1$ vertices, for some $k \geq 1$ —where X is identified with the larger class in the bipartition; see Fig. 1(c). Consider any non-empty matching in such an odd path. Traverse the path from one of its ends towards the other end, until you encounter the first matched vertex. If this vertex is in Y , then the previous vertex is in X and it is envious. If the first matched vertex is in X , then it is matched to the vertex after it in Y , and from that point onwards, every vertex in X must be matched to the vertex after it in Y in order not to envy. But the last vertex of the path is in X and has no vertex after it, so it is envious.

The examples above invoke the following questions.

- What characterises the graphs that contain a non-empty envy-free matching?
- Given a graph G , can an envy-free matching of maximum cardinality in G be found efficiently?

1.1. Envy-free matching and graph structure

We answer these questions by proving a structural theorem for bipartite graphs. We prove that in every bipartite graph, there is a unique partition of the vertices into two subsets—“good” and “bad”: the “good” subset is X -saturated (and thus contains the largest possible envy-free matching), while the “bad” subset has a structure similar to an odd path (and thus contains only an empty envy-free matching). The structure of this “bad” subset is defined formally below.

Definition 1.2. A bipartite graph $G := (X \cup Y, E)$ is called *Y -path-saturated* if, for some $k \geq 1$, there exist partitions $X = X_0 \cup \dots \cup X_k$ and $Y = Y_1 \cup \dots \cup Y_k$ where for all $i \geq 1$:

- There is a perfect matching between vertices of X_i and vertices of Y_i ;
- Every vertex in Y_i is adjacent to some vertex in X_{i-1} .

Every odd path with $|X| > |Y|$, as in Fig. 1(c), is Y -path-saturated. Fig. 1(d) shows another example of a Y -path-saturated graph; it can be seen that the structure of such a graph resembles that of an odd path. Every Y -path-saturated graph is Y -saturated, but the opposite is not true, as shown by Figs. 1(a,b).¹ In any Y -path-saturated graph, the only envy-free matching is \emptyset . The proof is similar to the one for odd paths above; we omit it since it is implied by Theorem 1.1(e) below. Thus the X -saturated graphs and the Y -path-saturated graphs are two extreme cases: the former contain the largest possible envy-free matching, while the latter contain only an empty envy-free matching. Our first result is that these two extremes are the building-blocks of *all* bipartite graphs. Below, $G[X', Y']$ denotes the subgraph of G induced by the vertices $X' \cup Y'$.

Theorem 1.1. Every bipartite graph $G = (X \cup Y, E)$ admits a unique partition $X = X_S \cup X_L$ and $Y = Y_S \cup Y_L$ satisfying the following three conditions:

- There are no edges between X_S and Y_L ;
 - The subgraph $G[X_S, Y_S]$ is Y -path-saturated;
 - The subgraph $G[X_L, Y_L]$ is Y -path-saturated;
- Moreover, this unique partition has the following additional properties:
- Every X_L -saturating matching in $G[X_L, Y_L]$ is an envy-free matching in G .
 - Every envy-free matching in G is contained in $G[X_L, Y_L]$.

For example, in Fig. 1(a), the entire graph is $G[X_L, Y_L]$, while $G[X_S, Y_S]$ is empty. In Fig. 1(b), $G[X_S, Y_S]$ contains the two leftmost edges and $G[X_L, Y_L]$ contains the rightmost (bold) edge, and there is one more edge between X_L and Y_S (but no edges between X_S and Y_L). In Fig. 1(c,d), the entire graph is $G[X_S, Y_S]$, while $G[X_L, Y_L]$ is empty.

¹ Note that the empty graph is Y -path-saturated (where $X_i = Y_j = \emptyset$ for all $i \geq 0, j \geq 1$). Also note that a Y -path-saturated graph may have isolated vertices (vertices with degree 0) in X —such vertices are contained in X_0 .

We call the unique partition of G , whose existence is guaranteed by [Theorem 1.1](#), the *EFM partition* of G .

As a corollary of [Theorem 1.1](#), one gets several useful conditions on a graph G admitting a non-empty envy-free matching. Two conditions are necessary and sufficient; the other is only sufficient. Below, $N_G(X')$ denotes the neighborhood of a subset $X' \subseteq X$ in G , i.e.: $N_G(X') := \{y' \in Y : \exists x' \in X' \text{ such that } (x', y') \in E\}$.

Corollary 1.1. *A bipartite graph $G := (X \dot{\cup} Y, E)$ admits a non-empty envy-free matching –*

- (a) *if and only if the bipartite graph $(X \dot{\cup} N_G(X), E)$ is not Y -path-saturated;*
- (b) *if $|N_G(X)| \geq |X| \geq 1$;*
- (c) *if and only if there is a subset $Y' \subseteq N_G(X)$ with $|N_G(Y')| \leq |Y'|$.*

Part (a) shows that all the “bad” graphs (graphs with only an empty envy-free matching) are similar to the odd-path example—they are all Y -path-saturated.

Parts (b) and (c) are similar to the condition in Hall’s marriage theorem [\[20\]](#). Hall’s theorem says that if (and only if) $|N_G(X')| \geq |X'|$ for any subset $X' \subseteq X$, then G admits an X -saturating matching. The strong condition of Hall is sufficient for the strong property of having an X -saturating matching; the weaker condition (b) is sufficient for the weaker property of having a non-empty envy-free matching. We note that part (b) was first proved by Luria [\[29\]](#); we present an alternative proof.

[Corollary 1.1\(b\)](#) can be slightly generalised to provide a lower bound on the cardinality of an envy-free matching.

Corollary 1.2. *Let $G := (X \dot{\cup} Y, E)$ be a bipartite graph with $|N_G(X)| \geq |X| \geq 1$. If, for some integer $k \geq 1$, every vertex in $N_G(X)$ has at least k neighbors in X , then G admits an envy-free matching of cardinality at least k .*

The structural theorem and its corollaries are proved in [Section 2](#).

Once all envy-free matchings are “captured” within a specific subgraph $G[X_L, Y_L]$, it is easy to develop optimisation algorithms for them. Below, the number of vertices in the smaller part of G is denoted by $\underline{n} := \min(|X|, |Y|)$, and the number of edges by $m := |E|$.

Theorem 1.2. *Given a bipartite graph $G = (X \dot{\cup} Y, E)$,*

- (a) *An envy-free matching of maximum cardinality in G can be found in $O(m\underline{n})$ time.*
- (b) *Given an edge cost function $w : E \rightarrow \mathbb{R}_{\geq 0}$, an envy-free matching of minimum total cost among those of maximum cardinality can be found within $O(m\underline{n} + \underline{n}^2 \log \underline{n})$ time.*

The algorithms are presented in [Section 3](#).

1.2. Envy-free matching in fair division

A *fair division problem* is a problem of allocating resources among people with different preferences, such that each person conceives his or her share as “fair” according to a given fairness criterion.

The algorithms of [Theorem 1.2](#) directly solve two variants of a problem known as *fair house assignment*. In this problem, the resources are indivisible, each agent must get at most a single resource, and the fairness criterion is envy-freeness. Part (a) solves a variant in which the goal is to maximise the number of agents assigned to a house that they like, subject to envy-freeness. Part (b) solves a variant in which each assignment of an agent to a house has a certain cost for society (e.g. the cost of building the house or of moving the agent to the house), and the goal is to minimise the total cost of the assignment, subject to envy-freeness and maximising the number of assigned agents. Both parts solve variants in which it is allowed to leave some houses unallocated.

Interestingly, the same algorithms, combined with [Corollary 1.1](#), can be used as subroutines in algorithms for various other fair division problems, both of divisible and of indivisible resources, in which *all* resources must be allocated. Each of these problems requires its own notation and definitions, which are presented formally in [Sections 4 and 5](#). Our results are presented informally below.

For a *divisible* resource (“cake”), we focus on a fairness criterion called *proportionality*, which means that each agent must get a piece that he/she values at least a fraction $1/n$ of the total cake value [\[38\]](#). There are various algorithms for proportional cake division, but most of them are not *symmetric*—the same agent might get a different value when playing first vs. playing second. This may lead to quarrels regarding who should play first. Chèze [\[12\]](#) presented a deterministic symmetric algorithm for proportional cake division, with an exponential run-time. He asked whether a polynomial-time algorithm exists. The following theorem answers his question; it is proved in [Section 4](#).

Theorem 1.3. *There is a deterministic, symmetric and polynomial-time algorithm that entirely allocates a divisible resource (“cake”) among n agents such that the value of each agent is at least $1/n$ of the total cake value.*

For *indivisible* objects, we focus on a fairness criterion called *1-out-of- k maximin-share*, which means that each agent weakly prefers his or her allocated bundle over the outcome of partitioning the objects into k subsets and getting the worst subset. Procaccia and Wang [32] proved that, when the objects are *goods* (i.e., each agent values each object at least 0), a 1-out-of- n maximin-share allocation may not exist for $n \geq 3$ agents. They asked whether a 1-out-of- $(n+1)$ maximin-share allocation exists. The following theorem makes a step towards an answer; it is proved in Section 5.

Theorem 1.4. *Given a set of indivisible goods, and n agents with additive valuations, there is a protocol that partitions all the goods among the agents, such that the value of each agent is at least the agent's 1-out-of- $(2n-2)$ maximin-share.*

The same algorithm can be used when the objects are *bads* (i.e., each agent values each object at most 0; such objects are also known as *chores*).

Theorem 1.5. *Given a set of indivisible bads, and n agents with additive valuations, there is a protocol that partitions all the bads among the agents, such that the value of each agent is at least the agent's 1-out-of- $\lfloor 2n/3 \rfloor$ maximin-share.*

Moreover, the same algorithm allows each agent to choose between other related fairness criteria, namely $(\ell-1)$ -out-of- $(\ell n-2)$ maximin-share for any $\ell \geq 2$, or $2/3$ -fraction maximin-share (see Appendix A). The main contribution of this paper thus lies not in solving a specific fair division problem, but rather in presenting a tool—envy-free matching—that can be applied as a subroutine in various kinds of fair division problems.

Some extensions of the basic model and some open questions are presented in Section 6.

Concepts similar to envy-free matching appeared in previous papers related to fair division, but they were hidden inside proofs of more specific algorithms [27,32,2,18,9]. Appendix B presents a detailed comparison. Presenting envy-free matching as a stand-alone graph-theoretic concept allows us to both simplify old algorithms and design new ones.

2. Bipartite graph structure and envy-free matchings

This section proves Theorem 1.1 and its corollaries. The main technical tool used is the *alternating sequence*.

2.1. Alternating sequences

Definition 2.1. Let M be a matching in a bipartite graph $G := (X \dot{\cup} Y, E)$. Let $X_0 \subset X$ be the subset of vertices unmatched by M . An M -alternating sequence starting at X_0 is a sequence of pairwise-disjoint subsets of vertices $X_0 - Y_1 - X_1 - Y_2 - X_2 - \dots$ where for all $i \geq 1$:²

- $Y_i = N_{G \setminus M}(X_{i-1}) \setminus (\cup_{j < i} Y_j)$;³
- $X_i = N_M(Y_i)$.

Given M and X_0 , it is simple to construct an M -alternating sequence starting at X_0 . Since the graph is finite, this construction eventually yields an empty subset—either X_i or Y_i for some i . Denote by $\mathcal{S}(M, X_0)$ the maximal M -alternating sequence starting at X_0 and ending before the first \emptyset . This $\mathcal{S}(M, X_0)$ induces a partition of the graph as follows (See Fig. 2):

- $X = X_S \dot{\cup} X_L$, where $X_S := \cup_{i \geq 0} X_i$ = the vertices of X participating in the sequence, and $X_L := X \setminus X_S$ = the Leftover vertices.
- $Y = Y_S \dot{\cup} Y_L$, where $Y_S := \cup_{i \geq 1} Y_i$ and $Y_L := Y \setminus Y_S$.

2.2. Alternating sequences and maximum-cardinality matchings

When M has maximum cardinality, its alternating sequences have useful properties.⁴

Lemma 2.1. *Let M be a maximum-cardinality matching in $G = (X \dot{\cup} Y, E)$ and $X_0 := X \setminus X_M$ = the subset of X unsaturated by M . Consider the partitions $X = X_S \dot{\cup} X_L$ and $Y = Y_S \dot{\cup} Y_L$ induced by the maximal alternating sequence $\mathcal{S}(M, X_0)$. Then:*

- (a) *There are no edges between X_S and Y_L ;*
- (b) *The subgraph $G[X_S, Y_S]$ is Y -path-saturated;*
- (c) *The subgraph $G[X_L, Y_L]$ is X -saturated.*

² The M -alternating sequence is closely related to the M -alternating path—a sequence of vertices $x_0 - y_1 - x_1 - \dots$ where each even edge is in M and each odd edge is not in M (or vice versa). The difference is that the elements in an M -alternating sequence are *subsets* of vertices rather than single vertices.

³ Here $G \setminus M$ denotes the graph G with the edges of M removed.

⁴ The vertices of X_L are exactly the vertices of X that are unreachable from X_0 in M -alternating paths. Thus X_L is reminiscent of the “unreachable” set in the Dulmage-Mendelsohn decomposition [25,33]. However, the reachability in the Dulmage-Mendelsohn decomposition is from the set of *all* unsaturated vertices, while the reachability in our case is only from X_0 —the set of unsaturated vertices in X .

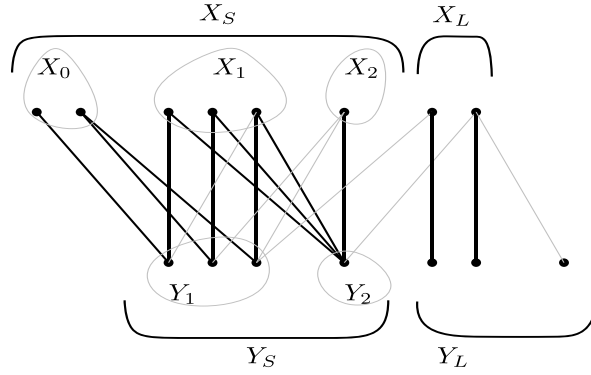


Fig. 2. A maximal M -alternating sequence $\mathcal{S}(M, X_0)$ and the induced partitions. The heavy vertical edges are edges of the underlying matching M . The heavy diagonal edges are edges of $E \setminus M$ used in constructing the alternating sequence. The light diagonal (gray) edges are edges of $E \setminus M$ not used in the construction.

Proof. Part (a). By construction, the set Y_S is exactly the set of neighbors of X_S in G .

Part (b). We first prove that $M[X_S, Y_S]$ —the subset of M contained in $G[X_S, Y_S]$ —saturates Y_S . Indeed, if, for some $i \geq 1$, some vertex $y_i \in Y_i$ were unmatched by M , then an M -alternating path could be traced along the edges used in the construction of $\mathcal{S}(M, X_0)$, namely: $y_i - X_{i-1} - Y_{i-1} - \dots - X_0$, where both end vertices are unmatched. By “inverting” the path, one could increase the size of the matching by one, but this contradicts the maximality of M . Hence, all vertices of $\bigcup_{i \geq 1} Y_i = Y_S$ are matched by M . By construction, the set of their matches in M is $\bigcup_{i \geq 1} X_i \subseteq X_S$.

This implies that the construction of $\mathcal{S}(M, X_0)$ ends at the X side, i.e., it ends at X_k for some $k \geq 0$. Now, the partitions $X_S = X_0 \cup \dots \cup X_k$ and $Y_S = Y_1 \cup \dots \cup Y_k$ satisfy the definition of a Y -path-saturated graph (Definition 1.2): for every $i \geq 1$, every vertex in Y_i is adjacent to some vertex in X_{i-1} , and there is a perfect matching between X_i and Y_i (along edges of M).

Part (c). We prove that $M[X_L, Y_L]$ —the subset of M contained in $G[X_L, Y_L]$ —saturates X_L . Indeed, by the lemma assumption, all vertices of X unmatched by M are contained in $X_0 \subseteq X_S$, so all vertices of X_L are matched by M . By construction, they must be matched to vertices not in any Y_i , so their matches must all lie inside Y_L . \square

Note that, in the special case in which G is X -saturated, the maximum matching M saturates X , so X_0 is empty and the M -alternating sequence is empty. In this case, $X_L = X$ and $Y_L = Y$. In the other extreme case, in which G is an odd path, X_0 always contains a single vertex which is one of the two endpoints, and the M -alternating sequence spans the entire graph, so $X_S = X$ and $Y_S = Y$.

2.3. Alternating sequences and envy-free matchings

The following lemma relates the three properties (a), (b), (c) above to envy-free matchings.

Lemma 2.2. Let $G = (X \cup Y, E)$, and consider any partitions $X = X_S \cup X_L$ and $Y = Y_S \cup Y_L$ satisfying properties (a), (b) and (c) of Lemma 2.1. Then:

(d) Every X_L -saturating matching in $G[X_L, Y_L]$ is an envy-free matching in G .

(e) Every envy-free matching in G is contained in $G[X_L, Y_L]$.

Note that Lemma 2.2 does not refer to a particular maximum matching—it holds for any partitions of X and Y that satisfy the properties (a), (b), (c) above.

Proof. [Proof of Lemma 2.2]

Part (d). Let W be an X_L -saturating matching in $G[X_L, Y_L]$. Since W saturates X_L , no vertex of X_L is envious. By property (a), there are no edges between X_S and Y_L . Since only vertices of Y_L are saturated by W , no vertex of X_S is envious. Hence, no vertex of X is envious, so W is an envy-free matching in G .

Part (e). Let W be any envy-free matching in G . Let $X_{SW} :=$ the subset of X_S saturated by W and $Y_{SW} :=$ the subset of Y_S saturated by W . We have to prove that both X_{SW} and Y_{SW} are empty. By property (a), vertices of X_{SW} can only be matched to vertices of Y_{SW} , so it is sufficient to prove that Y_{SW} is empty. The proof is by a counting argument. Let $k_{SW} := |Y_{SW}|$ and assume by contradiction that $k_{SW} > 0$.

By property (b), the graph $G[X_S, Y_S]$ is Y -path-saturated; denote the partitions appearing in Definition 1.2 by $X_S = X_0 \cup \dots \cup X_k$ and $Y_S = Y_1 \cup \dots \cup Y_k$. Let $i \geq 1$ be the smallest index such that a vertex of Y_i is matched by W , so that $Y_{SW} \subseteq \bigcup_{j \geq i} Y_j$. By Definition 1.2, all vertices of Y_{SW} are perfectly matched to vertices of $\bigcup_{j \geq i} X_j$; denote their matches by X'_{SW} . Note that $|X'_{SW}| = k_{SW}$. Every vertex $x \in X'_{SW}$ is adjacent (along an edge of the perfect matching) to a vertex of Y_{SW} , which is saturated by W . To ensure that x is not envious, x must be saturated by W too.

Let y' be a vertex in $Y_i \cap Y_{SW}$. By [Definition 1.2](#), it is adjacent to some vertex $x' \in X_{i-1}$. To ensure that x' is not envious, x' must be saturated by W too. But $x' \notin X'_{SW}$ since $X'_{SW} \subseteq \cup_{j \geq i} X_j$. Hence, there must be at least $k_{SW} + 1$ vertices of X_S that are saturated by W : the k_{SW} vertices of X'_{SW} , plus the vertex x' which is not in X'_{SW} . But this is a contradiction, since vertices of X_S can be matched only to vertices of Y_S , and only k_{SW} vertices of Y_S are matched by W . \square

We now have all the ingredients required to prove [Theorem 1.1](#), which we restate below.

2.4. The EFM partition

Proof. [Proof of [Theorem 1.1](#)] Let $G = (X \dot{\cup} Y, E)$ be a bipartite graph, M an arbitrary maximum matching in G , and $X_S \dot{\cup} X_L$ and $Y = Y_S \dot{\cup} Y_L$ the partitions induced by its maximal alternating sequence.

[Lemma 2.1](#) shows that these partitions satisfy properties (a), (b) and (c). [Lemma 2.2](#) then shows that parts (d) and (e) are satisfied too.

It remains to prove that the partitions are unique, that is, do not depend on the selection of the maximum matching M .

Consider alternative partitions $X = X'_S \dot{\cup} X'_L$ and $Y = Y'_S \dot{\cup} Y'_L$ satisfying properties (a), (b) and (c). Applying [Lemma 2.2\(d\)](#) to the partitions $X_S \dot{\cup} X_L, Y_S \dot{\cup} Y_L$ implies that there is an envy-free matching in G saturating X_L . Applying [Lemma 2.2\(e\)](#) to the partition $X'_S \dot{\cup} X'_L, Y'_S \dot{\cup} Y'_L$ implies that this matching must be contained in $G[X'_L, Y'_L]$; in particular, $X_L \subseteq X'_L$. Analogous arguments imply that $X'_L \subseteq X_L$. Hence $X'_L = X_L$. Hence also $X'_S = X_S$. Since $Y_S = N_G(X_S)$ and $Y'_S = N_G(X'_S)$, we also have $Y'_S = Y_S$. Hence also $Y'_L = Y_L$. \square

Note how the concept of envy-free matching helped us prove the uniqueness of the partition, which is a general fact about bipartite graphs.

2.5. Conditions for existence of envy-free matchings

We now prove [Corollary 1.1](#). It is simpler to prove in the following “reverse” formulation.

Corollary 2.1. [\equiv [Corollary 1.1](#)] A bipartite graph $G := (X \dot{\cup} Y, E)$ admits only an empty envy-free matching –

- (a) if and only if the bipartite graph $(X \dot{\cup} N_G(X), E)$ is Y -path-saturated;
- (b) only if $|N_G(X)| < |X|$ or $|X| = 0$;
- (c) if and only if $|N_G(Y')| > |Y'|$ for all non-empty subsets $Y' \subseteq N_G(X)$.

Proof. Part (a). Consider the unique partitions $X = X_S \dot{\cup} X_L$ and $Y = Y_S \dot{\cup} Y_L$ that exist by [Theorem 1.1](#). Parts (d,e) of this theorem imply that G admits only an empty envy-free matching iff X_L is empty. Hence it is sufficient to show that the graph $(X \dot{\cup} N_G(X), E)$ is Y -path-saturated iff X_L is empty.

If X_L is empty, then $X = X_S$ and $N_G(X) = Y_S$, so the graph $(X \dot{\cup} N_G(X), E)$ is Y -path-saturated.

Conversely, suppose $(X \dot{\cup} N_G(X), E)$ is Y -path-saturated, and define $X'_S := X$ and $X'_L := \emptyset$ and $Y'_S := N_G(X)$ and $Y'_L := Y \setminus N_G(X)$. Then, the partitions $X = X'_S \dot{\cup} X'_L$ and $Y = Y'_S \dot{\cup} Y'_L$ satisfy all three properties (a,b,c) of [Theorem 1.1](#): there are no edges between X'_S and Y'_L ; the subgraph $G[X'_S, Y'_S]$ is Y -path-saturated by assumption; and the subgraph $G[X'_L, Y'_L]$ is vacuously X -saturated. Now, the uniqueness of the partition implies that $X_L = X'_L = \emptyset$.

The next two parts follow from part (a):

Part (b). Every Y -path-saturated graph is either empty, or its Y side is smaller than its X side.

Part (c). In every Y -path-saturated graph, every non-empty subset in the Y side is contained in $\cup_{j \geq i} Y_j$ for some $i \geq 1$. It is perfectly matched to some subset of $\cup_{j \geq i} X_j$, and moreover, the vertices of Y_i are adjacent to some vertices of X_{i-1} ; therefore $|N_G(Y')| > |Y'|$.

Conversely, if $|N_G(Y')| > |Y'|$ for all non-empty $Y' \subseteq N_G(X)$, then every non-empty Y' that is perfectly matched to some subset $X' \subseteq X$, must be adjacent to some vertices in $X \setminus X'$. This implies that, in the unique EFM partition, Y_L must be empty – since otherwise it would have to be adjacent to some vertices in X_S , in contradiction to property (a). \square

Corollary 2.2. [\equiv [Corollary 1.2](#)] Let $G := (X \dot{\cup} Y, E)$ be a bipartite graph with $|N_G(X)| \geq |X| \geq 1$. If, for some integer $k \geq 1$, every vertex in $N_G(X)$ has at least k neighbors, then G admits an envy-free matching of cardinality at least k .

Proof. [Corollary 1.1\(b\)](#) implies that G admits a non-empty envy-free matching. Any matched vertex in Y has at least k neighbors. By envy-freeness, all these neighbors must be matched. \square

3. Algorithms for finding envy-free matchings

This section applies the structural results of the previous section to prove [Theorem 1.2](#).

The proof uses a simple algorithm (Algorithm 1) for finding the unique EFM partition of a bipartite graph G . The algorithm first finds an arbitrary maximum-cardinality matching in G ; this can be done using the classic algorithm of Hopcroft and Karp [21]. Ramshaw and Tarjan [34] show that this algorithm runs within $O(mn)$ time.

Then, the algorithm finds the set X_0 of unmatched vertices and the maximal alternating sequence $\mathcal{S}(M, X_0)$. The sets X_S, Y_S are just the union of the subsets of X, Y participating in the sequence, and the sets X_L, Y_L are the remaining vertices in X, Y . These can all be found in time linear in $n + m$, since the algorithm entails to examine every vertex and scan its adjacent edges a constant number of times. Therefore, the total run-time of Algorithm 1 is $O(mn)$.

3.1. Maximum cardinality envy-free matching

Proof. [Proof of [Theorem 1.2\(a\)](#)] The theorem claims that, given a bipartite graph G , an envy-free matching of maximum cardinality can be found within $O(mn)$ time.

This can be done simply by the following algorithm:

1. Find the EFM partition of G using Algorithm 1.
2. Return an arbitrary maximum-cardinality matching in $G[X_L, Y_L]$.

By [Theorem 1.1\(a\)](#), the returned matching saturates X_L ; by [Theorem 1.1\(d\)](#), it is envy-free; by [Theorem 1.1\(e\)](#), no other envy-free matching can saturate any vertex of X_S . Therefore, the returned matching is indeed a maximum-cardinality envy-free matching. \square

To save time, instead of returning an arbitrary maximum-cardinality matching in $G[X_L, Y_L]$, we can re-use the maximum-cardinality matching M which is needed for computing the EFM partition, and return its subset $M[X_L, Y_L]$ —the set of edges of the matching M that link vertices of X_L to vertices of Y_L ; see Algorithm 2. By [Lemmas 2.1 and 2.2](#), this subset is an envy-free matching and it saturates X_L .

Algorithm 1: Finding the EFM partition of a bipartite graph

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

Output: The unique EFM partition of G to $X_S \cup X_L = X$ and $Y_S \cup Y_L = Y$.

- 1: Find a maximum-cardinality matching M in G .
 - 2: Let $X_0 := X \setminus X_M$ = the vertices of X unmatched by M .
 - 3: Compute the maximal M -alternating sequence $\mathcal{S}(M, X_0)$.
 - 4: Return $X_S := \cup_{i \geq 0} X_i$ and $X_L := X \setminus X_S$ and $Y_S := \cup_{i \geq 1} Y_i$ and $Y_L := Y \setminus Y_S$.
-

Algorithm 2: Finding an envy-free matching of maximum cardinality

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

Output: An envy-free matching of maximum cardinality in G .

- 1: Find a maximum-cardinality matching M in G .
 - 2: Find the EFM partition $X = X_S \cup X_L$ and $Y = Y_S \cup Y_L$ using Algorithm 1 (steps 2–4).
 - 3: Return the sub-matching $M[X_L, Y_L]$.
-

Example 3.1. Consider the bipartite graph in [Fig. 2](#). Algorithm 1 finds the EFM partition $X = X_S \cup X_L$ and $Y = Y_S \cup Y_L$ as indicated in the figure. Assuming the matching M is denoted by bold vertical edges, Algorithm 2 returns the set of two rightmost vertical edges, which is an envy-free matching of size 2.

3.2. Minimum cost envy-free matching

Proof. [Proof of Theorem 1.2(b)] The theorem assumes that the graph $G := (X \cup Y, E)$ is endowed with an edge-cost function $w : E \rightarrow \mathbb{R}_{\geq 0}$. The cost of a matching $M \subseteq G$ is defined as $w(M) := \sum_{e \in M} w(e)$. The theorem claims that an envy-free matching of minimum total cost among those of maximum cardinality can be found within $O(m\bar{n} + \bar{n}^2 \log \bar{n})$ time.

The proof uses Algorithm 3. Just like Algorithm 2, it starts by finding the unique EFM partition of G . The difference is in the last step: instead of returning $M[X_L, Y_L]$, which is an arbitrary X_L -saturating matching, it returns an X_L -saturating matching of minimum cost.

Finding a minimum-cost maximum-cardinality matching is known as the *assignment problem*. Since X_L and Y_L may be of different sizes, it is an *unbalanced assignment problem*. Ramshaw and Tarjan [34] provide a comprehensive survey of algorithms for the unbalanced assignment problem. In particular, they show that the famous Hungarian method can be generalised to unbalanced bipartite graphs, and its run-time is $O(m \cdot \bar{n} + \bar{n}^2 \log \bar{n})$.

By Theorem 1.1, all envy-free matchings in G are contained in the subgraph $G[X_L, Y_L]$, and all maximum-cardinality matchings in $G[X_L, Y_L]$ saturate X_L and are therefore envy-free. Hence, the Hungarian method on $G[X_L, Y_L]$ yields a maximum-cardinality envy-free matching of minimum cost in G . Since $|X_L| \leq |Y_L|$, the run-time of the Hungarian method is $O(m \cdot |X_L| + |X_L|^2 \log |X_L|)$ which is in $O(m \cdot \bar{n} + \bar{n}^2 \log \bar{n})$. \square

Algorithm 3: Finding a minimum-cost maximum-cardinality envy-free matching

Input: A bipartite graph $G := (X \cup Y, E)$ and a cost function $w : E \rightarrow \mathbb{R}_{\geq 0}$.

Output: An envy-free matching of minimum total cost.

- 1: Find the EFM partition of G into $X = X_S \cup X_L$ and $Y = Y_S \cup Y_L$ using Algorithm 1.
 - 2: Find and return a minimum-cost maximum-cardinality matching in $G[X_L, Y_L]$.
-

Example 3.2. Consider again Fig. 2. The subgraph $G[X_L, Y_L]$ contains three edges; denote them from left to right by e_1, e_2, e_3 . This subgraph contains two maximum-cardinality matchings: $\{e_1, e_2\}$ and $\{e_1, e_3\}$. Both are envy-free matchings in G . Algorithm 3 returns one of them, depending on whether $w(e_1) + w(e_2)$ or $w(e_1) + w(e_3)$ is smaller.

4. Application to fair division

4.1. Generic fair division problem

We consider first the following generic fair division problem.

- There is a set C , representing a resource that has to be divided among n agents.
- For each agent $i \in [n]$ there is a measure (an additive set function) $V_i : 2^C \rightarrow \mathbb{R}$, representing the agent's valuation of different parts of the resource.
- For each agent i , there is a threshold value $t_i \in \mathbb{R}$.

A *t-fair division* of C is a partition of C into n subsets, $C = Z_1 \cup \dots \cup Z_n$, such that

$$\forall i \in [n] : V_i(Z_i) \geq t_i.$$

The existence of a *t-fair division* depends on the threshold values t_i and on the nature of the resource C . For example, if $t_i > V_i(C)/n$ for all $i \in [n]$, then a *t-fair division* obviously might not exist (e.g. when the agents' valuations are identical). Similarly, if C contains a single (indivisible) object, and the threshold values are positive, then a *t-fair division* does not exist. We prove that a *t-fair division* exists whenever the threshold values are “reasonable”, in the sense defined below.

Definition 4.1 (*Reasonable threshold*). Given a resource C , a value measure V_i on C and an integer $n \geq 2$, a real number $t_i \in \mathbb{R}$ is called a *reasonable threshold* for V_i if:

- (1) There exists a partition of C into $C_1 \cup \dots \cup C_n$, such that

$$\forall j \in [n] : V_i(C_j) \geq t_i.$$

(Informally, i can partition C into n subsets that are acceptable by i 's own standards).

- (2) For every $k \in \{1, \dots, n-1\}$, and every k disjoint subsets $U_1, \dots, U_k \subseteq C$, if

$$\forall j \in [k] : V_i(U_j) < t_i,$$

then there exists a partition of $C \setminus \bigcup_{j \in [k]} U_j$ into $C_1 \dot{\cup} \dots \dot{\cup} C_{n-k}$, such that

$$\forall j \in [n-k] : V_i(C_j) \geq t_i.$$

(Informally, if any k Unacceptable subsets are given away, then i can partition the remainder into $n-k$ acceptable subsets).⁵

Theorem 4.1. Consider a resource C and value measures V_1, \dots, V_n on C . If t_i is a reasonable threshold for V_i for all $i \in [n]$, then a t -fair division exists.

Proof. The proof is constructive and uses Algorithm 4, which generalises an algorithm of Kuhn [27]. It is described in detail below.

Step 1 asks some arbitrary agent a to partition the resource into $|X|$ pieces that are acceptable by her own standards. In the first iteration, this is possible thanks to condition (1) in the definition of a reasonable threshold; below, we will show that this is possible in the following iterations too.

Step 2 constructs a bipartite graph where each agent is adjacent to all the pieces that are acceptable for him.

Step 3 finds a maximum-cardinality envy-free matching in this graph. Since a is adjacent to all $|X|$ pieces, $|N_G(X)| \geq |X|$, so by Corollary 1.1, a non-empty envy-free matching is found. Each matched agent $i \in X_M$ receives a piece with a value of at least t_i , so the fairness condition is satisfied for these agents.

Step 4 removes the matched agents and pieces, and goes back to step 1 to handle the remaining agents. Let k be the total number of pieces allocated in all previous iterations, so that $|X \setminus X_M| = n - k$. By the definition of envy-free matching, for each unmatched agent $i \in X \setminus X_M$, the value of each allocated piece is less than t_i . Therefore, by condition (2) in the definition of a reasonable threshold, each remaining agent can partition the remaining resource as required in step 1.

The size of X decreases by at least 1 in each iteration. Therefore, after at most n iterations the algorithm ends with a t -fair division. \square

Algorithm 4: The Lone Divider algorithm. Based on Kuhn [27].

Input:

- A set C representing a resource to divide.
- A set $X = [n]$ of agents with measures $(V_i)_{i=1}^n$ on C .
- Reasonable threshold values $(t_i)_{i=1}^n$ (see Definition 4.1).

Output: A partition $C = Z_1 \dot{\cup} \dots \dot{\cup} Z_n$ such that $V_i(Z_i) \geq t_i$ for all $i \in [n]$.

- 1: Choose an arbitrary agent $a \in X$. Partition C into $|X|$ disjoint subsets, $(C_j)_{j \in X}$, with $V_a(C_j) \geq t_a$ for all $j \in X$.
 - 2: Define a bipartite graph G with the agents of X on the one side and the set $Y := \{C_1, \dots, C_{|X|}\}$ on the other side. Add an edge (i, C_j) whenever $V_i(C_j) \geq t_i$.
 - 3: Using Algorithm 2, find a maximum-cardinality envy-free matching M in G .
Give each matched element in Y_M to the agent paired to it in X_M .
 - 4: Let $X \leftarrow X \setminus X_M$ = the unallocated agents and $C \leftarrow C \setminus (\cup Y_M)$ = the unallocated resource. If $X \neq \emptyset$, go back to step 1.
-

Remark 4.1. At each iteration, the partition in step 1 requires to ask agent a at most n queries. Constructing the graph in step 2 requires to ask each of the other agents at most n queries (one query for each piece). There are at most n iterations, so each agent is asked $O(n^2)$ queries, and the total number of required queries is $O(n^3)$.

Remark 4.2. The Lone Divider algorithm is not *strategyproof*—agent i might gain from reporting a false value measure V_i . This holds even when $n = 2$, when Lone Divider is equivalent to cut-and-choose. See e.g. Brams and Taylor [10] for a discussion of this issue. In this paper, we ignore these strategic issues and assume that the agents report their valuations truthfully.

Below we apply Theorem 4.1 to some specific fair division problems.

4.2. Proportional cake-cutting

A *proportional cake-cutting* problem is a special case of the generic fair division problem, in which —.

⁵ Condition (1) is the special case of condition (2) for $k = 0$; it is presented as a different condition for the sake of clarity.

- The resource C is continuous; it is usually called “cake” and represented by the real interval $[0, 1]$.
- The value measures $(V_i)_{i \in [n]}$ are nonatomic.
- For each agent i , the threshold value is $t_i = V_i(C)/n$.

These threshold values are reasonable (see [Definition 4.1](#)):

Condition (1) holds thanks to the assumption that the value measures are nonatomic. For each measure V_i , the cake can be partitioned into n subsets of equal measure, which is exactly $V_i(C)/n$.

Condition (2) holds since, if we remove from C any k subsets with a value smaller than $V_i(C)/n$, then the value of the remainder is at least $V_i(C) - \frac{k}{n} V_i(C) = (n - k) \cdot V_i(C)/n$; hence it can be partitioned into $n - k$ subsets of value at least $V_i(C)/n$.

Hence, by [Theorem 4.1](#), the Lone Divider algorithm can be used to find a proportional cake-cutting. In fact, the Lone Divider algorithm was originally stated specifically for the proportional cake-cutting problem. Steinhaus presented it for 3 agents. Kuhn [27] extended it to an arbitrary number of agents. The cases $n = 3$ and $n = 4$ are described in detail by Brams and Taylor [10][pages 31–35], and the general case is described in detail by Robertson and Webb [35][pages 83–87]. Note how the use of envy-free matchings lets us present this algorithm in a much shorter way.

The Lone Divider algorithm requires $O(n^3)$ queries, as there are at most n iterations, and in each iteration, each of the n agents should evaluate at most n pieces. Another cake-cutting algorithm, by Even and Paz [15], requires only $O(n \log n)$ queries. However, Lone Divider has other advantages. One advantage is that it does not assume that all valuations are positive, or even that all valuations have the same sign: it is applicable to a “mixed manna” setting, in which each part of the cake may be positive to some agents and negative to others [9,3]. A second advantage of Lone Divider is that can be modified to be not only fair but also *symmetric*. This is explained in the following subsection.

4.3. Symmetric algorithm for proportional cake-cutting

This subsection proves [Theorem 1.3](#) regarding a symmetric proportional cake-cutting algorithm. A fair division algorithm is called *symmetric* if the value each agent receives depends only on the valuations of the agents, and not on the order in which the algorithm processes them. In other words, if we run the algorithm, permute the agents, and run the algorithm again, every agent has the same value in both runs. Most cake-cutting algorithms are not symmetric. For example, in Algorithm 4, the cutter in step 1 (agent a) always receives exactly $1/n$ of the total value, while other agents may get more than $1/n$.⁶ This might make the agents quarrel over who the cutter will be. One solution is to select the cutter uniformly at random. But is there a symmetric *deterministic* algorithm?

Manabe and Okamoto [30] presented deterministic symmetric algorithms for two and three agents. The case $n \geq 4$ remained open until Chèze [12] presented a deterministic symmetric algorithm for any number of agents. Chèze mentions that the number of arithmetic operations required by his algorithm may be exponential in n , and asks whether there exists a deterministic symmetric algorithm in which the number of arithmetic operations required is polynomial in n . We answer his question in the affirmative by combining his algorithm with our Algorithm 3 for minimum-cost envy-free matching. The combined algorithm is shown as Algorithm 5. The general scheme is similar to the Lone Divider method, but there are several important changes, which are explained below.

Algorithm 5: A symmetric algorithm for proportional cake-cutting. Based on Chèze [12].

Input: A cake $C = [0, 1]$, and a set $X = [n]$ of agents with nonatomic measures $(V_i)_{i=1}^n$ on C . The measures are normalised such that $\forall i \in [n] : V_i(C) = n$.

Output: A partition $C = Z_1 \dot{\cup} \dots \dot{\cup} Z_n$ such that $\forall i \in [n] : V_i(Z_i) \geq 1$.

- 1: Ask each agent $i \in X$ to produce a vector z_i of $n - 1$ marks, $0 < z_{i,1} < \dots < z_{i,n-1} < 1$, which partition C into n sub-intervals with a value of exactly 1 for i .
- 2: Choose from the mark vectors $\{z_i | i \in [n]\}$ the one that is smallest in lexicographic order. Cut C by these marks. Denote the resulting pieces by C_1, \dots, C_n .
- 3: Define a bipartite graph G with the agents of X on the one side and the pieces (denoted by Y) on the other side. Add an edge (i, C_j) whenever $V_i(C_j) \geq 1$.
- 4: Assign to each agent $i \in X$ a weight $w(i) \in 0 \dots n - 1$ that is a function of the set of neighbors $N_G(i)$, such that two agents $i, i' \in X$ have the same weight iff $N_G(i) = N_G(i')$.
- 5: Assign to each piece $C_j \in Y$ a unique weight $w(C_j) \in 0 \dots n - 1$, such that the weight of the leftmost piece (adjacent to

(continued on next page)

⁶ It is assumed here that the agents answer the queries truthfully, based on their real value measure.

- 0) is 0, the weight of the next-leftmost piece is 1, etc.
- 6: Assign to each edge $(i, C_j) \in E$ the cost $w(i, C_j) := 2^{n \cdot w(i) + w(C_j)}$. Using Algorithm 3 find a minimum-cost maximum-cardinality envy-free matching M in G .
- 7: Let $X_Y \subseteq X$ be the set of all agents that are adjacent to all pieces in Y (i.e., the agents $i \in X$ with $N_G(i) = Y$). Give each agent in X_Y an arbitrary piece from $N_M(X_Y)$.
- 8: Let $X_R := X_M \setminus X_Y$. Partition the agents in X_R to subsets according to their weight. I.e., for some $k \geq 1$, find a partition $X_R = X_1 \cup \dots \cup X_k$, such that i, i' belong to the same X_j iff $w(i) = w(i')$.
- 9: For all $j \in [k]$, recursively divide the union of pieces in $N_M(X_j)$ among the agents in X_j .
- 10: Recursively divide the union of pieces in $Y \setminus Y_M$ among the agents in $X \setminus X_M$ (by going back to step 1).

The first change from Lone Divider is that the initial partition should be decided in a way that depends only on the valuations. This is done in step 2 using lexicographic ordering.

For example, if Alice cuts the cake at 0.3, 0.7, Bob cuts at 0.4, 0.6 and Carl at 0.2, 0.6, then the algorithm selects Carl partition, since the cut-pair (0.2, 0.6) is lexicographically smaller than the other two cut-pairs. Hence, in the initial partition, the cake is cut at 0.2 and 0.6, so $C_1 = [0, 0.2]$ and $C_2 = [0.2, 0.6]$ and $C_3 = [0.6, 1]$.

The second change is in steps 4–6. Since there may be many different envy-free matchings, one of them must be selected in a way that depends only on the valuations. One way to select a unique envy-free matching is to assign to each edge, a cost that is a unique power of two. This guarantees that each subset of edges has a unique cost, so there is a unique minimum-weight maximum-cardinality envy-free matching. However, symmetry requires that the edge costs themselves should depend only on the valuations. Therefore, edges may have different costs only if they are adjacent to different pieces (since the pieces depend only on the valuations), or to agents with different valuations. This motivates the weighting scheme in steps 4–6. An example of a graph with some edge costs is shown in Fig. 3. Note that the length of the costs in binary is polynomial in the graph size.

In the special case that each agent has a unique set of neighbors, the agent weights are unique, the edge costs are unique powers of two, each matching has a unique cost, and thus the minimum-cost envy-free matching M found in step 6 is uniquely determined by the valuations. In this special case, the algorithm can just proceed as in Algorithm 4: give each piece in Y_M to the agent matched to it in X_M , and recursively divide the remaining cake—the union of pieces in $Y \setminus Y_M$ —among the remaining agents in $X \setminus X_M$.

In the general case, there may be several different minimum-cost envy-free matchings, and step 6 returns one of them, in a way that may depend on the agents' order. Therefore, to preserve symmetry, care must be taken to ensure that the agents' values are not sensitive to the minimum-cost matching selected. Note that all these minimum-cost matchings have the same set X_M of matched agents—it is exactly the set X_L defined by the unique partition of Theorem 1.1. Moreover, by the determination of edge costs, all these matchings have the same set Y_M of matched pieces. So the sets X_M and Y_M are uniquely determined by the valuations; only the pairing of agents in X_M with pieces in Y_M is not uniquely determined and must be handled in the following steps.

Consider first the set X_Y defined in step 7. Note that it contains at least one agent—the agent responsible to the lexicographically-smallest partition selected in step 2 (in Fig. 3, the set X_Y contains the two agents with weight 1). By envy-freeness of M , all agents in X_Y are matched by M . By the determination of edge costs, all minimum-cost envy-free matchings M' in G have the same set $N_{M'}(X_Y)$, i.e., in all these matchings, the same pieces are allocated to the agents in

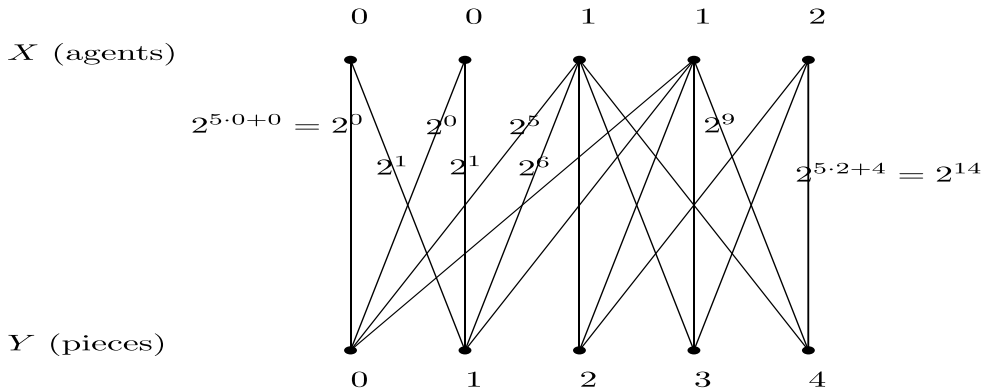


Fig. 3. Bipartite agent-piece graph with weights determined by Algorithm 5. The numbers below the pieces in Y and above the agents in X are their weights. Each piece has a unique weight, while each agent has a weight that is a 1-to-1 function of its set of neighbors. The two leftmost agents have the same set of neighbors so they have the same weight (0); the two agents adjacent to them have the same set of neighbors so they have the same weight (1); the rightmost agent has a different set of neighbors. The numbers on the edges are their costs (due to space constraints, only some of the costs are written).

X_Y . Since all agents in X_Y value all pieces in Y at exactly 1, it is possible to give each agent in X_Y an arbitrary piece in $N_M(X_Y)$, for example, based on the agents' indices. This arbitrary choice does not affect the value of any agent; all agents in X_Y are treated symmetrically.

Consider now the sets X_1, \dots, X_k defined in step 8 (in Fig. 3 there are two such sets: the set X_1 contains the two leftmost agents whose weight is 0, and the set X_2 contains the rightmost agent whose weight is 2). For each $j \in [k]$, all agents in X_j have the same weight, so they have the same set of neighbors. Hence, by envy-freeness of M , if one agent in a set X_j is matched by M , then all agents in X_j must be matched by M too. By the determination of edge costs, all minimum-cost envy-free matchings M' in G have the same set $N_{M'}(X_j)$, i.e., in all these matchings, the same pieces are allocated to the agents in X_j . Here, it is not possible to give each agent in X_j an arbitrary piece in $N_M(X_j)$, since each agent in X_j may value the pieces in $N_M(X_j)$ differently. However, by definition of the graph G , all agents in X_j value each piece in $N_M(X_j)$ at least 1, so they value the union of $N_M(X_j)$ at least $|X_j|$. Therefore, by recursively dividing the union of $N_M(X_j)$ among the agents in X_j , each agent in X_j is guaranteed a value of at least 1. All agents in X_j are treated symmetrically.

Finally, step 10 of Algorithm 5 is analogous to step 4 of Algorithm 4: by envy-freeness of M , the agents in $X \setminus X_M$ value each piece given away at less than 1, so they value the remaining cake at more than $|X \setminus X_M|$, so the recursive call gives each of them a value of at least 1.

From the above discussion, it follows that Algorithm 5 is symmetric, it runs in polynomial time, and it finds a proportional cake-allocation, as claimed in Theorem 1.3.

4.4. Other applications

Another application of envy-free matching is found in a generalisation of cake-cutting called *multi-cake cutting*. In this problem, the cake is made of m pairwise-disjoint sub-cakes ("islands"), and each agent should be given a piece that overlaps at most k islands, for some fixed integer $k \geq 1$. When $k < m$, it may be impossible to guarantee to each agent $1/n$ of the total value. A natural question is what fraction *can* be guaranteed, as a function of k, m, n . Recently, Segal-Halevi [37] proved that the fraction is $\min\left(\frac{1}{n}, \frac{k}{m+n-1}\right)$. The proof is constructive and uses envy-free matching in a different way than the Lone Divider algorithm. Envy-free matchings were also applied for cutting a cake in the form of a general graph, representing e.g. a road network [14].

5. Fair allocation of discrete objects

A *fair object allocation* problem is a special case of the generic fair division problem, in which –

- The resource C is a finite set; its elements are called *objects* or *items*.
- The value measures $(V_i)_{i \in [n]}$ are any additive set functions $V_i : 2^C \rightarrow \mathbb{R}$.

Objects with a positive value to all agents are usually called *goods*; objects with a negative value are called *bads* or *chores*.

In this setting, a proportional allocation might not exist, i.e., it may be impossible to find a t -fair division with threshold values $t_i = V_i(C)/n$; consider for example the case in which C contains a single object. Hence, proportionality is often relaxed to the *maximin share*, which is defined below.

5.1. The maximin share

For every agent $i \in [n]$ and integers $1 \leq \ell \leq d$, the ℓ -out-of- d maximin-share of i from C , denoted $MMS_i^{\ell\text{-out-of-}d}(C)$, is defined as

$$MMS_i^{\ell\text{-out-of-}d}(C) := \max_{\mathbf{P} \in \text{Partitions}(C,d)} \min_{Z \in \text{Union}(\mathbf{P},\ell)} V_i(Z)$$

where the maximum is over all partitions of C into d subsets, and the minimum is over all unions of ℓ subsets from the partition. Informally, $MMS_i^{\ell\text{-out-of-}d}(C)$ is the largest value that agent i can get by partitioning C into d piles and getting the worst ℓ piles. Obviously $MMS_i^{\ell\text{-out-of-}d}(C) \leq \frac{\ell}{d} V_i(C)$, and equality holds iff C can be partitioned into d subsets with the same value. Thus, $MMS_i^{\ell\text{-out-of-}d}(C)$ can be thought of as $\frac{\ell}{d} V_i(C)$ "rounded down to the nearest object". The maximin share with $\ell = 1$ was introduced by Budish [11]. The generalisation to arbitrary $\ell \geq 1$ was done by Babaioff et al. [5,6].

The maximin-share is well-defined both for goods and for bads. For example, suppose C contains three goods o_1, o_2, o_3 , and some agent i values them at 2, 3, 4 respectively. Then $MMS_i^{1\text{-out-of-}2}(C) = 4$, by the partition $\{o_1, o_2\}, \{o_3\}$. If the objects are bads and their values are $-2, -3, -4$, then $MMS_i^{1\text{-out-of-}2}(C) = -5$ by the same partition.

Note that for goods $MMS_i^{1\text{-out-of-}k}(C)$ is a weakly-decreasing function of k , while for bads the opposite is true—it is a weakly-increasing function of k .

The values $t_i = \text{MMS}_i^{1-\text{out-of-}n}(C)$ are particularly interesting, since they are the largest values that satisfy condition (1) for reasonable thresholds. When $n = 2$, these values satisfy condition (2) too (note that we only have to check the case $k = 1$): if one subset with value less than $\text{MMS}_i^{1-\text{out-of-}2}(C) \leq V_i(C)/2$ is removed, then the value of the remaining objects is more than $V_i(C)/2 \geq \text{MMS}_i^{1-\text{out-of-}2}(C)$. Therefore, a t -fair allocation exists.

Procaccia and Wang [32] prove that, for any $n \geq 3$, there might not exist a t -fair division with $t_i = \text{MMS}_i^{1-\text{out-of-}n}(C)$. They present a *multiplicative approximation* to the threshold values, $t_i = \gamma \cdot \text{MMS}_i^{1-\text{out-of-}n}(C)$, for some fraction $\gamma \in (0, 1)$. They present an algorithm attaining a t -fair division for a fraction γ that equals $3/4$ for $n \in \{3, 4\}$, and approaches $2/3$ as $n \rightarrow \infty$.

An alternative approximation, suggested by Budish [11], is $t_i = \text{MMS}_i^{1-\text{out-of-}n+1}(C)$. In contrast to the multiplicative approximation, the existence of a t -fair allocation with these thresholds depends only on the agents' rankings of the bundles, and not on the specific values assigned to them. In other words, an allocation that is t -fair with the 1-out-of- $(n+1)$ MMS thresholds remains fair even if the agents' value functions are modified, as long as the *order* between the bundles' values remains the same for every agent. Therefore, we call this kind of approximation an *ordinal approximation*. Regarding the 1-out-of- $(n+1)$ MMS, Procaccia and Wang [32] 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 $n = 3$.⁷ Below we prove that the Lone Divider algorithm can be used to attain a t -fair division with $t_i = \text{MMS}_i^{1-\text{out-of-}2n-2}(C)$, which for $n = 3$ coincides with $\text{MMS}_i^{1-\text{out-of-}n+1}(C)$.

5.2. Maximin-share allocation of goods

The proof of Theorem 1.4 uses the following combinatorial lemmas.⁸

Lemma 5.1. Let $(a_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $a_j \in [0, 1]$. If $\sum_j a_j \geq A$ for some integer A , then the a_j can be partitioned into $\lceil A/2 \rceil$ subsets such that the sum of each subset is at least 1.

Proof. Collect the a_j sequentially into subsets, starting with a_1 , until the sum of the current subset is at least one. Continue constructing subsets in this way until all the a_j -s are arranged in subsets. Let $s+1$ be the number of constructed subsets, where the sum of the first s subsets is at least 1 and the sum of the last subset (which may be empty) is less than 1. Since $a_j \leq 1$, the sum of each of the first s subsets is less than 2. Therefore, the sum of all subsets is less than $2s+1$. Hence, $2s+1 > A$ so $2s \geq A$ so $s \geq \lceil A/2 \rceil$, since s and A are integers. \square

Lemma 5.2. Let $(b_j)_{j=1}^N, (c_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $b_j \in [0, c_j]$ and $c_j \geq 1$. If $\sum_j b_j \geq \left(\sum_j c_j\right) - k$ for some integer $k \geq 0$, then the b_j can be partitioned into $\lceil (N-k)/2 \rceil$ subsets such that the sum of each subset is at least 1.

Figuratively, the lemma says the following. There are N bottles of water, each of which contains at least 1 litre. Some water is spilled out of some of the bottles, such that the total amount spilled out is at most k litres. Then, the bottles can be grouped into $\lceil (N-k)/2 \rceil$ subsets, such that the bottles in each subset together contain at least 1 litre of water.

Proof. [Proof of Lemma 5.2] Let $a_j := b_j/c_j$ for all $j \in [N]$. Then $a_j \in [0, 1]$, and

$$\begin{aligned} \sum_{j=1}^N a_j &= \sum_{j=1}^N \frac{b_j}{c_j} = N - \sum_{j=1}^N \frac{c_j - b_j}{c_j} \geq N - \sum_{j=1}^N (c_j - b_j) \\ &\quad \text{since } c_j \geq 1 \\ &= N - \left(\sum_j c_j - \sum_j b_j \right) \geq N - k \\ &\quad \text{since } \sum_j c_j - \sum_j b_j \leq k. \end{aligned}$$

By Lemma 5.1, the a_j can be partitioned into $\lceil (N-k)/2 \rceil$ subsets with a sum of at least 1. Since $b_j = c_j \cdot a_j \geq a_j$, the sum of b_j corresponding to the a_j in each subset is at least 1. \square

We now prove Theorem 1.4, which says that there always exists an allocation of goods among n agents giving each agent i a value of at least $\text{MMS}_i^{1-\text{out-of-}2n-2}(C)$.

⁷ Perhaps they wanted to write it in the margin but the margin was too narrow ☹

⁸ We are grateful to user bof of MathOverflow.com for the proof idea: <https://mathoverflow.net/a/334754/34461>.

Proof. [Proof of Theorem 1.4] We prove that the threshold values $t_i = \text{MMS}_i^{1\text{-out-of-}2n-2}(C)$ are reasonable, as defined in Definition 4.1. Condition (1) is obviously satisfied: by definition of MMS, each agent i can partition C into n subsets worth at least $\text{MMS}_i^{1\text{-out-of-}n}(C)$, which is at least as large as $\text{MMS}_i^{1\text{-out-of-}2n-2}(C)$.

For Condition (2), let $N := 2n - 2$, and let $(C_j)_{j \in [N]}$ be a 1-out-of- N MMS partition of agent i . Let $c_j := V_i(C_j)/t_i$. By definition of the MMS, $c_j \geq 1$ for all $j \in [N]$.

Suppose we remove some objects whose total value is at most $k \cdot t_i$. Let $b_j :=$ the total value remaining in C_j after the removal, divided by t_i . So $b_j \in [0, c_j]$, and $\sum_j b_j \geq (\sum_j c_j) - k$. By Lemma 5.2, the b_j can be partitioned into $\lceil (N - k)/2 \rceil$ subsets with a sum of at least 1. This corresponds to a partition of the remaining objects into $\lceil (N - k)/2 \rceil$ bundles with a value of at least t_i . Since $\lceil (N - k)/2 \rceil = (n - 1) - \lfloor k/2 \rfloor \geq n - k$ whenever $k \geq 1$, condition (2) holds, and by Theorem 4.1, the Lone Divider algorithm finds a t -fair division. \square

Remark 5.1. As mentioned above, the Lone Divider algorithm requires $O(n^3)$ queries. However, with indivisible objects, answering each query requires agent i to compute the 1-out-of- $(2n - 2)$ MMS. This requires solving an instance of the multi-way number partitioning problem, which is known to be NP-hard. If the number of agents and objects is sufficiently small, then the problem can be solved optimally by heuristic algorithms [36]. Otherwise, a PTAS of Woeginger [39] can be used to find in polynomial time, for each $\epsilon > 0$, a partition in which the value of each part is at least $(1 - \epsilon)\text{MMS}_i^{1\text{-out-of-}2n-2}(C)$. Then, the Lone Divider algorithm can be executed with $t_i = (1 - \epsilon)\text{MMS}_i^{1\text{-out-of-}2n-2}(C)$.

Remark 5.2. The Lone Divider algorithm cannot guarantee the 1-out-of- $(2n - 3)$ MMS. As an example, suppose there are $4n - 6$ goods. Suppose some agent Alice values some $2n - 3$ goods at $1 - \epsilon$ and the others at ϵ , so her 1-out-of- $(2n - 3)$ MMS equals 1. It is possible that the first divider partitions the goods such that all the $2n - 3$ low-value goods are in a single bundle, and this bundle is allocated to another agent in the envy-free matching. If, in the next round, Alice is the divider, then she cannot partition the remaining $2n - 3$ high-value goods into $n - 1$ bundles with a value of at least 1. Recently, Hosseini et al. [22] developed a modified Lone Divider algorithm, that attains a better approximation.

5.3. Maximin-share allocation of bads

The proof of Theorem 1.5 uses the following combinatorial lemma.

Lemma 5.3. Let $(a_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $a_j \in [0, 1]$. If $\sum_j a_j \leq A$ for some integer A , then the a_j can be partitioned into $2A + 1$ subsets such that the sum of each subset is at most 1.

Proof. Collect the a_j sequentially into subsets, starting with a_1 , until the sum of the current subset is at least one. Continue constructing subsets in this way until all the a_j -s are arranged in subsets. Let $s + 1$ be the number of constructed subsets, where the sum of the first s subsets is at least 1 and the sum of the last subset (which may be empty) is less than 1. The sum of all subsets is at least s , so $s \leq A$. From each subset, remove the last element added to it. Since $a_j \leq 1$, we now have $2s + 1$ subsets each of which has a sum of at most 1. By adding empty subsets if needed, we get $2A + 1$ subsets with a sum of at most 1. \square

Lemma 5.4. Let $(b_j)_{j=1}^N, (c_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $b_j \in [0, c_j]$ and $c_j \in [0, 1]$. If $\sum_j b_j \leq (\sum_j c_j) - k$ for some integer $k \geq 0$, then the b_j can be partitioned into $2(N - k) + 1$ subsets such that the sum of each subset is at most 1.

Figuratively, the lemma says the following. There are N bottles of water, each of which contains at most 1 litre. Some water is spilled out of some of the bottles, such that the total amount spilled out is at least k litres. Then, the bottles can be grouped into $2(N - k) + 1$ subsets, such that the bottles in each subset together contain at most 1 litre.

Proof. [Proof of Lemma 5.4] Let $a_j := b_j/c_j$ for all $j \in [N]$. Then $a_j \in [0, 1]$, and

$$\begin{aligned}\sum_{j=1}^N a_j &= \sum_{j=1}^N \frac{b_j}{c_j} = N - \sum_{j=1}^N \frac{c_j - b_j}{c_j} \leq N - \sum_{j=1}^N (c_j - b_j) \quad \text{since } c_j \leq 1 \\ &= N - \left(\sum_j c_j - \sum_j b_j \right) \leq N - k \quad \text{since } \sum_j c_j - \sum_j b_j \geq k.\end{aligned}$$

By Lemma 5.3, the a_j can be partitioned into $2(N - k) + 1$ subsets with a sum of at most 1. Since $b_j = c_j \cdot a_j \leq a_j$, the sum of b_j corresponding to the a_j in each subset is at most 1. \square

We now prove Theorem 1.5, which says that there always exists an allocation of bads among n agents giving each agent i a value of at least $\text{MMS}_i^{1\text{-out-of-}N}(C)$, where $N = \lfloor 2n/3 \rfloor$.

Proof. [Proof of Theorem 1.5] We prove that the threshold values $t_i = \text{MMS}_i^{1\text{-out-of-}N}(C)$ are reasonable. Condition (1) is obviously satisfied: by definition of MMS, each agent i can partition C into n subsets worth $\text{MMS}_i^{1\text{-out-of-}n}(C)$. Since the values of all objects are negative, this value is at least $\text{MMS}_i^{1\text{-out-of-}N}(C)$ for any $N \leq n$ (when the bads are partitioned into more subsets, the value in each subset is larger).

For condition (2), let $(C_j)_{j \in [N]}$ be a 1-out-of- N MMS partition of agent i . By definition of the MMS, $V_i(C_j) \geq t_i$. The condition holds trivially whenever $k \leq n - N$, since in this case $n - k \geq N$, so adding $(n - k) - N$ empty bundles gives $n - k$ bundles with value at least t_i (note that $t_i \leq 0$). Therefore, we assume now that $k \geq n - N + 1$.

Let $c_j := V_i(C_j)/t_i$. Since both $V_i(C_j)$ and t_i are negative, $0 \leq c_j \leq 1$ for all $j \in [N]$. Suppose we remove some objects whose total value is at most $k \cdot t_i$. Let $b_j :=$ the total value remaining in C_j after the removal, divided by t_i . So $b_j \in [0, c_j]$, and $\sum_j b_j \leq (\sum_j c_j) - k$ (again the sign of inequality is reversed since both quantities are negative). By Lemma 5.4, the b_j can be partitioned into $2(N - k) + 1$ subsets with a sum of at most 1. This corresponds to a partition of the remaining bads into bundles of value at least t_i . By the definition of N ,

$$\begin{aligned}3N &\leq 2n \\ \Rightarrow 2N + 1 &\leq 2n - N + 1 = n + (n - N + 1) \\ \Rightarrow 2N + 1 &\leq n + k && \text{whenever } k \geq n - N + 1 \\ \Rightarrow 2(N - k) + 1 &\leq n - k.\end{aligned}$$

By adding empty bundles if needed, agent i can partition the remaining bads into $n - k$ bundles with a value of at least t_i . Therefore, condition (2) holds, and by Theorem 4.1, the Lone Divider algorithm finds a t -fair division. \square

Remark 5.3. Suppose n is divisible by 3 and $N = 2n/3$. Then the Lone Divider algorithm cannot guarantee the 1-out-of- $(N + 1)$ MMS. Suppose the bads' values for Alice are

- $N + 1$ big bads with value $-1/2 - 1/2V$ each, for $V = N + \frac{n}{3} + 1$;
- $N + 1$ sets of $V - 1$ small bads with value $-1/2V$ each. So the total value of each set is $-1/2 + 1/2V$, and the total number of small bads is $(N + 1)(V - 1) = NV + \frac{n}{3}$.

The bads can be grouped into $N + 1$ sets with value -1 , so $\text{MMS}_A^{1\text{-out-of-}N+1}(C) = -1$. But it is possible that the first envy-free matching allocates $k = n - N$ unacceptable bundles, each of which contains $2V + 1$ small bads (with total value $-1 - 1/2V$); note that the total number of small bads allocated is $(n - N)(2V + 1) = \frac{n}{3}(2V + 1) = NV + \frac{n}{3}$. Then N agents remain, and there are $N + 1$ big bads that Alice cannot partition into N bundles with value at least -1 .

5.4. Other maximin-share guarantees

The Lone Divider algorithm can find a t -fair division with various other threshold values (see Appendix A). For example, it can guarantee to each agent his 2-out-of- $(3n - 2)$ MMS. For some agents, this guarantee may be better than 1-out-of- $(2n - 2)$ MMS. For example, with $n = 3$, if C contains 7 objects and agent i values all of them at 1, then $\text{MMS}_i^{2\text{-out-of-}7}(C) = 2$ while $\text{MMS}_i^{1\text{-out-of-}4}(C) = 1$. More generally, for every integer $\ell \geq 2$, it is possible to find a t -fair division with $t_i = \text{MMS}_i^{\ell-1\text{-out-of-}\ell n-2}(C)$. The algorithm can also guarantee a multiplicative approximation of $t_i = \frac{2n}{3n-1} \cdot \text{MMS}_i^{1\text{-out-of-}n}(C) > \frac{2}{3} \text{MMS}_i^{1\text{-out-of-}n}(C)$. For proving all these variants, it is sufficient to prove that the threshold vectors are reasonable, using combinatorial lemmas analogous to Lemmas 5.1 and 5.2; see Appendix A for details.

Algorithm 4 can even make different guarantees to different agents. For example, it is possible to set $t_1 = \text{MMS}_1^{1\text{-out-of-}2n-2}(C)$ and $t_2 = \text{MMS}_2^{2\text{-out-of-}3n-2}(C)$ and $t_3 = \frac{2}{3} \text{MMS}_3^{1\text{-out-of-}n}(C)$. Finally, if some agents are computationally-bounded, and cannot calculate their MMS partition exactly, they can use an approximation algorithm like that of Woeginger [39] to calculate an approximate MMS partition—a partition in which the value of each bundle is at least

$(1 - \epsilon) \text{MMS}_i^{1\text{-out-of-}2n-2}(C)$, for some $\epsilon > 0$. They can then participate in Algorithm 4 with $t_i = (1 - \epsilon) \text{MMS}_i^{1\text{-out-of-}2n-2}(C)$. The algorithm then guarantees to these agents a value of at least their approximate MMS. This does not affect the guarantee to computationally-unbounded agents, who are still guaranteed at least their exact MMS.

While there are now algorithms that attain better multiplicative approximation factors for goods [2,7,18,17] and for bads [7,24], it may be useful to have a simple algorithm that allows each agent to choose between a multiplicative and various ordinal approximations.⁹

In general, the Lone Divider method cannot guarantee a 1-out-of- $(n + 1)$ MMS allocation (see Remark 5.2). Corollary 1.2 can help to identify special cases in which such allocations do exist. First, recall that, if all n agents have the same valuation function, then by definition a 1-out-of- n MMS allocation exists. The same is true if all n agents have the same MMS partition (even if their valuations are different). Moreover, the same is true even if only $n - 1$ agents have the same MMS partition, since then it is possible to let the n -th agent pick a bundle and divide the remaining $n - 1$ bundles among the remaining $n - 1$ agents. The following theorem generalizes this observation.

Theorem 5.1. *Let $\ell \geq 1$ be an integer. Given a set of indivisible goods, if there exists a partition P of the goods into n bundles, in which $n - \ell$ agents value each bundle at least as their 1-out-of- $(n + \ell - 1)$ MMS, then there exists a 1-out-of- $(n + \ell - 1)$ MMS allocation. In particular, existence is ensured if $n - \ell$ agents have identical valuations.*

Proof. [Proof of Theorem 5.1] Let $N := n + \ell - 1$. Apply the Lone Divider algorithm with $t_i = \text{MMS}_i^{1\text{-out-of-}N}(C)$ for all $i \in [n]$, using in Step 1 the partition P from the theorem statement. In Step 2, by assumption, each bundle in Y is adjacent to at least $n - \ell$ agents in X . Hence, by Corollary 1.2, the number of matched agents is at least $n - \ell$. Denote this number by k . Lemma 5.2 implies that each of the remaining $n - k$ agents can partition the remaining objects into $\lceil (N - k)/2 \rceil$ bundles worth at least t_i . The assumption $k \geq n - \ell$ implies that

$$\begin{aligned} \lceil (N - k)/2 \rceil &= \lceil (n + \ell - 1 - k)/2 \rceil \\ &\geq \lceil (n + (n - k) - 1 - k)/2 \rceil \\ &= \lceil (2n - 2k - 1)/2 \rceil \\ &= n - k. \end{aligned}$$

Hence, we can proceed with Algorithm 4 and get a t -fair division. \square

As an example, for $n = 4$, a 1-out-of-5 MMS allocation exists whenever there exists a partition in which some two agents value each bundle by at least their 1-out-of-5 MMS; particularly, when some two agents have identical valuations. The general case remains open.

6. Extensions and open problems

6.1. Symmetric envy and non-bipartite graphs

Our definition of an envy-free matching is asymmetric in that it considers the envy of vertices in X only. For example, the odd path in Fig. 1(c) has a non-empty EFM w.r.t. Y but not w.r.t. X .

One can define a matching as *symmetric-envy-free* if any unmatched vertex in G is not adjacent to any matched vertex in G . This definition extends naturally to non-bipartite graphs.

With this symmetric definition, the algorithmic problems studied here become much easier (and less interesting). Suppose first that $G = (V, E)$ is a connected graph (bipartite or not). If some matching in G saturates some vertex $v \in V$ but does not saturate some other vertex $v' \in V$, then on the path between v and v' , at least one vertex is envious. Therefore, a symmetric envy-free matching saturates either all vertices or no vertices. Hence, a connected graph admits a non-empty symmetric-envy-free matching if and only if it admits a perfect matching.

Therefore, an arbitrary graph admits a non-empty symmetric-envy-free matching if and only if it has a connected component admitting a perfect matching. A maximum cardinality (minimum cost) symmetric-envy-free matching is just the union of all perfect matchings (of minimum cost) of such connected components.

6.2. Star matchings

The envy-freeness concept can be generalised from a matching (a set of vertex-disjoint edges) to an *r-star matching*—a set of vertex-disjoint copies of the star $K_{1,r}$, where the star center is in X and the star leaves are in Y . An *envy-free r-star matching* is then an *r-star matching* in which every vertex in X that is not matched (as a center), is disconnected from any vertex in Y that is matched (as a leaf). Our results can be easily generalised to *r-star matchings*. For example, the following theorem generalises Theorem 1.2a) and Corollary 1.1(b).

⁹ Recently, Bogomolnaia et al. [9] have shown that an algorithm very similar to Algorithm 4 attains a different approximate-fairness notion that they call “Pro1” (proportionality up to at most one object).

Theorem 6.1. For every integer $r \geq 1$,

- (a) There is a polynomial-time algorithm that, given any bipartite graph $G = (X \cup Y, E)$, finds a maximum-cardinality envy-free r -star matching in G .
- (b) If $|N_G(X)| \geq r|X| \geq 1$, then G admits a non-empty envy-free r -star matching.

Proof. Given G , construct an auxiliary bipartite graph $G' := (X' \cup Y, E')$, where X' has r clones of every vertex in X , and E' has an edge from each clone v^x of $x \in X$ to every vertex $y \in N_G(x)$. For $x \in X$, let v_1^x, \dots, v_r^x denote the r clones of x in X' . There is a many-to-one correspondence between envy-free matchings in G' and envy-free r -star matchings in G :

- (1) Consider any envy-free matching W' in G' . For all $x \in X$, consider the subgraph

$$G'[\{v_1^x, \dots, v_r^x\}, N_G(x)].$$

This subgraph forms a complete bipartite graph. Hence, if $(v_i^x, y) \in W'$ for some $i \in [r]$, then W' being envy-free in G' means that W' must saturate all of $\{v_1^x, \dots, v_r^x\}$. All these vertices can only be paired to vertices of $N_G(x)$; all edges thus used by W' (in G') correspond to different edges in G whose one end is x (the ends in $N_G(x)$ are distinct). Collapsing every set $\{v_1^x, \dots, v_r^x\}$ saturated by W' back to its origin x in G gives an envy-free r -star matching in G .

- (2) Given an envy-free r -star matching W in G , create a matching W' in G' by connecting, for each saturated vertex $x \in X$, each clone v_i^x of x to one of the r vertices in Y matched to x in W . Note that, for each saturated vertex $x \in X$, there are $r!$ ways to connect the clones of x to its neighbors, so there are many different matchings W' corresponding to W . However, all such matchings have the same cardinality, and every such matching is envy-free in G' : for every vertex x that is saturated by W , all its clones are saturated by W' and thus are not envious; for every vertex x that is unmatched by W , all its clones are not adjacent to any matched vertex in Y , and thus are not envious either.

We now use the above correspondence for proving the two claims in the theorem.

- (a) The size (number of edges) of the matchings W' in G' is exactly r times the size (number of stars) of the corresponding matching W in G . Hence, applying Algorithm 2 to G' yields a maximum-cardinality r -star matching in G .
- (b) If $|N_G(X)| \geq r|X| \geq 1$, then $|N_{G'}(X')| \geq |X'| \geq r \geq 1$, so G' satisfies the premise of [Corollary 1.1b](#)) and thus admits a non-empty envy-free matching W' . It corresponds to an envy-free r -star matching W in G .

□

In an r -star matching, each vertex in X is matched to either 0 or r vertices in Y . One can also consider allocation problems in which each vertex in X may be connected to *any number* in $\{0, \dots, r\}$ of vertices in Y (but each vertex in Y may still be connected to at most one vertex in X). A many-to-one matching $M \subseteq E$ is called *envy-free* if for every two vertices $x_1, x_2 \in X$, the number of neighbors of x_1 matched to x_1 is at least as large as the number of neighbors of x_1 matched to x_2 . This definition reduces to [Definition 1.1](#) when $r = 1$.

When $r = \infty$, the problem of finding an envy-free many-to-one matching is equivalent to the problem of *fair allocation with binary additive valuations*. Y is a set of discrete goods, and X is a set of agents. Each agent values each object at either 0 or 1, and values each set of objects as the sum of the values of its elements. The goal is to allocate the objects among the agents such that each agent values its own bundle at least as much as the bundle of any other agent. Aziz et al. [4] proved that deciding whether an envy-free allocation of all objects in Y exists is NP-complete (remark after Theorem 11; the same result was proved in a different way by Hosseini et al. [23] at Proposition 3). Therefore, the problem of finding an envy-free one-to-many matching of maximum cardinality is NP-hard. However, the reductions consider only allocations in which all objects are allocated—they do not allow partial allocations.¹⁰ Therefore, the following problem remains open.

Is there a polynomial-time algorithm for deciding whether a given bipartite graph admits a non-empty envy-free one-to-many matching?

6.3. Maximum value envy-free matching

Suppose that the edge weights are interpreted as *values* rather than costs, and thus one is interested in finding an envy-free matching of *maximum* total value.

¹⁰ For example, in the reduction of Hosseini et al. [23], Property 2 does not necessarily hold for partial allocations: it is possible that each edge-agent receives a single edge-good, each dummy-agent receives nothing, and the vertex-goods remain unallocated. This is a non-empty envy-free allocation that does not correspond to an equitable coloring of G .

The unbalanced Hungarian method can be easily adapted to find an X -saturating matching of maximum value [34]. Hence, Algorithm 3 can be adapted to find a maximum-cardinality envy-free matching of maximum value. The following lemma shows that, whenever all values are non-negative, the maximum value of *any* envy-free matching is always attained by some *maximum-cardinality* envy-free matching.¹¹

Lemma 6.1. *Let $G := (X \cup Y, E)$ be a bipartite graph. Let W be an envy-free matching in G . Then, W is contained in some maximum-cardinality envy-free matching in G .*

Proof. Let M be a maximum-cardinality envy-free matching in G . By Theorem 1.1, this M is contained in $G[X_L, Y_L]$ and saturates X_L . For each vertex $x \in X_L$, denote by $N_M(x)$ the vertex in Y_L matched to it by M . Let $M' := \{(x, N_M(x)) : x \in X_L, x \text{ is unsaturated by } W\}$,

Envy-freeness of W implies that, for any vertex $x \in X_L$ unsaturated by W , $N_M(x)$ must be unsaturated by W too. Hence, $W \cup M'$ is a matching. It saturates X_L , so by Theorem 1.1 it is a maximum-cardinality envy-free matching in G . \square

Hence, by taking Algorithm 3 and replacing “minimum-cost” by “maximum-value”, one gets an algorithm for finding a maximum-value envy-free matching.

However, this algorithm is meaningful only when $w(x, y)$ represent the value of the pairing (x, y) to “society” as a whole (or to the social planner), since it appears in the maximisation objective but not in the envy definition. An alternative interpretation is that $w(x, y)$ represents the subjective value of y to x . This interpretation leads to a different definition of envy-freeness. Given a function w on the edges and a matching M , define:

$$w(x, M) = \begin{cases} w(x, y) & \text{If } x, y \in M; \\ 0 & \text{If } x \text{ is unmatched by } M. \end{cases}$$

A matching $M \subseteq E$ is called *w-envy-free* if, for every vertex $x \in X$ and every matched vertex $y' \in Y_M : w(x, M) \geq w(x, y')$, i.e., every agent in X weakly prefers his or her own house (if any) to any house assigned to another agent. This definition reduces to Definition 1.1 when all edges have the same weight. The problem of finding a *w-envy-free* matching was studied by several authors in parallel to the present work:

- Gan et al. [16] present a polynomial-time algorithm for finding an X -saturating *w-envy-free* matching, if and only if such a matching exists.
- Beynier et al. [8] consider a similar problem in a more complex setting where agents are located on a network, and each agent only envies his or her neighbors in the network.
- Kamiyama et al. [26] study the problem of finding an X -saturating matching that is not necessarily *w-envy-free*, but it maximizes the number of vertices of X for which the *w-envy-free* condition is satisfied. They prove that this problem is NP-hard even for binary weights (all weights are either 0 or 1). Moreover, for general weights, the problem is hard to approximate under some common complexity-theoretic assumptions.

In contrast to our work, these three works do not consider partial matchings (matchings that do not necessarily saturate X). To illustrate the difference between the settings, suppose $|X| = |Y| = n$, and

$$w(x_i, y_j) = \begin{cases} 2 & 1 \leq i \leq n, j = n; \\ 1 & i = j < n. \end{cases}$$

In the unique X -saturating matching, x_i is matched to y_i for all $i \in [n]$, and *w-envy-freeness* is satisfied only for x_n . However, the matching in which x_i is matched to y_i for all $i \in [n - 1]$ and x_n remains unmatched is a partial *w-envy-free* matching of size $n - 1$.

Is there a polynomial-time algorithm that, for any value function w , finds a partial *w-envy-free* matching of maximum cardinality? Of maximum value?

Remark 6.1. Irving et al. [25] present an algorithm for finding a *rank-maximal* matching—a matching that maximises the number of agents who are matched to their first choice, then the number of agents who are matched to their second choice, etc. A matching that is rank-maximal among the set of all envy-free matchings can be found by an algorithm similar to Algorithm 3: in the last step, instead of solving the unbalanced assignment problem, apply the algorithm of Irving et al. [25] to $G[X_L, Y_L]$.

6.4. Relaxations of envy-free matching

Since non-empty envy-free matchings might not exist, one may be interested in relaxations. For example, given a real $\alpha > 0$, a matching M is called *α -fraction envy-free* if for every unmatched $x \in X \setminus X_M$: $|N_G(x) \cap Y_M| \leq \alpha |N_G(x)|$. That is, agents

¹¹ Note that the above does not hold for arbitrary matchings.

unsaturated by M are willing to “tolerate” at most an α -fraction of their acceptable houses being assigned to someone else. Alternatively, given an integer $c \geq 0$, M is called c -additive envy-free if for every unmatched $x \in X \setminus X_M$: $|N_G(x) \cap Y_M| \leq c$.

Is there a polynomial-time algorithm for finding a maximum-cardinality matching among the approximate-envy-free matchings, for any of the above approximation notions?

From a probabilistic perspective, 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. Dickerson et al. [13] and Manurangsi and Suksompong [31].

Declaration of Competing Interest

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

Acknowledgments

Erel acknowledges Zur Luria [29], who first provided an existential proof to Corollary 1.1(b), as well as instructive answers by Yuval Filmus, Thomas Klimpel and bof in MathOverflow.com, Max, Vincent Tam and Elmex80s in MathStackExchange.com, and helpful comments by anonymous referees to the WTAF 2019 workshop, the EC 2020 conference, and the Information Sciences journal. This research is partly supported by Israel Science Foundation grant 712/20.

Appendix A. Variants of maximin share fairness

This appendix shows various fairness guarantees that can be attained by the Lone Divider algorithm when allocating discrete goods.

A.1. Ordinal approximation

The first fairness guarantee uses several lemmas.

Lemma A.1. Let $\ell \geq 2$ be an integer. Let $(a_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $a_j \in [0, 1]$. If $\sum_j a_j \geq A$ for some integer A , then the a_j can be partitioned into $\lfloor (A+1)/\ell \rfloor$ subsets such that the sum of each subset is at least $\ell - 1$.

Note that Lemma 5.1 corresponds to the special case $\ell = 2$.

Proof. Collect the a_j sequentially into subsets, starting with a_1 , until the sum of the current subset is at least $\ell - 1$. Continue constructing subsets in this way until all the a_j -s are arranged in subsets. Let $s+1$ be the number of constructed subsets, where the sum of the first s subsets is at least $\ell - 1$ and the sum of the last subset (which may be empty) is less than $\ell - 1$. Since $a_j \leq 1$, the sum of each of the first s subsets is less than ℓ . Therefore, the sum of all subsets is less than $\ell s + \ell - 1$. Hence, $\ell s + \ell - 1 > A$ so $\ell(s+1) > A+1$ so $s+1 > \lfloor (A+1)/\ell \rfloor$ so $s \geq \lfloor (A+1)/\ell \rfloor$.

Lemma A.2. Let $L \geq 1$ be an integer. Let $(d_j)_{j=1}^N, (c_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $d_j \in [0, 1]$, and the sum of every L -tuple of c_j is at least L . If $\sum_j d_j \geq L$, then $\sum_j c_j \cdot d_j \geq \sum_j d_j$ (in particular $\sum_j d_j \geq L$ implies $\sum_j c_j \cdot d_j \geq L$ and $\sum_j d_j > L$ implies $\sum_j c_j \cdot d_j > L$).

Note that in the special case $L = 1$, $c_j \geq 1$ for all $j \in [N]$, so the claim is trivial.

Proof. Let $D \geq \subset [0, 1]^N$ be the set of vectors \mathbf{d} satisfying the lemma condition, i.e., $\sum_j d_j \geq L$. Let $D \geq \subset D \geq$ be those vectors satisfying $\sum_j d_j = L$. The lemma can be stated as a dot product: $\mathbf{c} \cdot \mathbf{d} \geq L$. We first show that it holds for all $\mathbf{d} \in D \geq$ and then for all $\mathbf{d} \in D \geq$.

Consider a vector $\mathbf{d} \in D \geq$. If it has only integer coordinates, then it must have exactly L ones and $N - L$ zeros, so the sum $\mathbf{c} \cdot \mathbf{d} = \sum_j c_j \cdot d_j$ contains exactly L elements from \mathbf{c} . By assumption, their sum is at least L , so the lemma holds.

Otherwise, \mathbf{d} has a non-integer coordinate, say $d_{j_1} \in (0, 1)$. Since the sum of coordinates is an integer—it must have another non-integer coordinate, say $d_{j_2} \in (0, 1)$. Given $\epsilon > 0$, define a (j_1, j_2, ϵ) -shift of \mathbf{d} as a vector \mathbf{d}' given by

$$d'_j = \begin{cases} d_{j_1} - \epsilon & j = j_1 \\ d_{j_2} + \epsilon & j = j_2 \\ d_j & \text{otherwise} \end{cases}$$

Choosing $\epsilon = \min(d_{j_1}, 1 - d_{j_2})$ guarantees that $\mathbf{d}' \in D_+$ and it has fewer non-integer coordinates than \mathbf{d} (either $d'_{j_1} = 0$ and $d'_{j_2} = d_{j_1} + d_{j_2}$, or $d'_{j_2} = 1$ and $d'_{j_1} = d_{j_1} + d_{j_2} - 1$). Choosing j_1, j_2 such that $c_{j_1} \geq c_{j_2}$ guarantees that each such shift decreases the dot product,

$$\mathbf{c} \cdot \mathbf{d}' = \mathbf{c} \cdot \mathbf{d} - c_{j_1} \cdot \epsilon + c_{j_2} \cdot \epsilon \leq \mathbf{c} \cdot \mathbf{d}$$

There is a finite sequence of (j_1, j_2, ϵ) -shifts culminating in a vector $\mathbf{d}'' \in D_+$ with only integer coordinates. Therefore,

$$\mathbf{c} \cdot \mathbf{d} \geq \mathbf{c} \cdot \mathbf{d}'' \geq L,$$

so the claim holds for all $\mathbf{d} \in D_+$.

For a vector $\mathbf{d} \in D_-$, Let $\mathbf{d}' := \mathbf{d} \cdot \frac{L}{\sum_j d_j}$; by assumption $\mathbf{d}' \in D_+$, so $\mathbf{c} \cdot \mathbf{d}' \geq L$. Now,

$$\mathbf{c} \cdot \mathbf{d} = \frac{\sum_j d_j}{L} \mathbf{c} \cdot \mathbf{d}' = \sum_j d_j \cdot \frac{\mathbf{c} \cdot \mathbf{d}'}{L} \geq \sum_j d_j \geq L,$$

so the claim holds for all $\mathbf{d} \in D_-$ too. \square

Lemma A.3. Let $\ell \geq 2$ be an integer. Let $(b_j)_{j=1}^N, (c_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $b_j \in [0, c_j]$, and the sum of every $(\ell - 1)$ -tuple of c_j is at least $\ell - 1$. If $\sum_j b_j \geq (\sum_j c_j) - (\ell - 1) \cdot k$ for some integer $k \geq 0$, then the b_j can be partitioned into $\lfloor (N - (\ell - 1)k + 1)/\ell \rfloor$ subsets such that the sum of each subset is at least $\ell - 1$.

Figuratively, the lemma says the following. There are N bottles of water, such that each $(\ell - 1)$ -tuple of bottles contains at least $\ell - 1$ litres. Some water is spilled out of some of the bottles, such that the total amount spilled out is at most $(\ell - 1) \cdot k$ litres. Then, the bottles can be grouped into $\lfloor (N - (\ell - 1)k + 1)/\ell \rfloor$ subsets, such that each subset contains at least $\ell - 1$ litres of water. Note that Lemma 5.2 corresponds to the special case $\ell = 2$.

Proof. [Proof of Lemma A.3] Let $a_j := b_j/c_j$ for all $j \in [N]$. Then $a_j \in [0, 1]$, and

$$\sum_{j=1}^N a_j = \sum_{j=1}^N \frac{b_j}{c_j} = N - \sum_{j=1}^N \frac{c_j - b_j}{c_j}.$$

Let $d_j := \frac{c_j - b_j}{c_j}$. By assumption,

$$\sum_j c_j \cdot d_j = \sum_j c_j - \sum_j b_j \leq (\ell - 1)k.$$

Applying Lemma A.2 in the contrapositive direction with $L := (\ell - 1)k$ implies that

$$\sum_j d_j \leq (\ell - 1)k$$

too. Therefore,

$$\sum_{j=1}^N a_j = N - \sum_j d_j \geq N - (\ell - 1)k.$$

By Lemma A.1, the a_j can be partitioned into $\lfloor (N - (\ell - 1)k + 1)/\ell \rfloor$ subsets with a sum of at least $\ell - 1$. In each such subset G ,

$$\sum_{j \in G} a_j \geq \ell - 1.$$

Applying Lemma A.2 with $L := \ell - 1$ and $d_j := a_j$ yields

$$\sum_{j \in G} c_j \cdot a_j \geq \ell - 1 \Rightarrow \sum_{j \in G} b_j \geq \ell - 1.$$

Therefore, there exists a partition of the b_j as claimed. \square

Corollary A.4. For every $\ell \geq 2$, the Lone Divider algorithm can attain a t -fair division of goods with $t_i = \text{MMS}_i^{(\ell-1)\text{-out-of-}(n-2)}(C)$.

Proof. It is sufficient to prove that the t_i are reasonable thresholds for each agent i . We verify condition (2) for every integer $k \geq 0$. Let $N := \ell n - 2$, and let $(C_j)_{j \in [N]}$ be a partition of C attaining the maximum in the definition of $\text{MMS}_i^{\ell-1-\text{out-of-}\ell n-2}(C)$. By definition of the MMS, the sum of every $(\ell - 1)$ -tuple of $V_i(C_j)$ is at least t_i . Let $c_j := (\ell - 1) \cdot V_i(C_j)/t_i$; so the sum of every $(\ell - 1)$ -tuple of c_j is at least $\ell - 1$.

Suppose we remove some objects whose total value is at most $k \cdot t_i$. Let $b_j :=$ the value of objects remaining at C_j , after the removal, multiplied by $(\ell - 1)/t_i$; so $b_j \in [0, c_j]$, and $\sum_j b_j \geq (\sum_j c_j) - (\ell - 1) \cdot k$. Lemma A.3 implies that the b_j can be partitioned into subsets with a value of at least $\ell - 1$. This corresponds to a partition of the remaining goods into bundles with a value of at least t_i . The number of such bundles is at least

$$\begin{aligned} \lfloor (N - (\ell - 1)k + 1)/\ell \rfloor &= \left\lfloor \frac{\ell n - 1 - (\ell - 1)k}{\ell} \right\rfloor = \\ &= \left\lfloor n - k + \frac{k - 1}{\ell} \right\rfloor \\ &\geq n - k. \end{aligned}$$

So condition (2) holds, and by Theorem 4.1, Lone Divider finds a t -fair division. \square

A.2. Multiplicative approximation

The Lone Divider algorithm can also provide a multiplicative approximation, similarly to the “APX-MMS” algorithm of Amanatidis et al. [2]. The proof uses the following lemma.

Lemma A.5. Let $(b_j)_{j=1}^N, (c_j)_{j=1}^N$ be real numbers such that for all $j \in [N]$: $b_j \in [0, c_j]$ and $c_j \geq 1$. If $\sum_j b_j \geq \sum_j c_j - \frac{2n}{3n-1}k$ for some integer $k \geq 0$, then the b_j can be partitioned into $n - k$ subsets such that the sum of each subset is at least $\frac{2n}{3n-1}$.

Proof. Define

- d_0 —the number of j such that $(c_j - b_j)/c_j \in [0, \frac{n-1}{3n-1}]$.
- d_1 —the number of j such that $(c_j - b_j)/c_j \in (\frac{n-1}{3n-1}, \frac{2n-1}{3n-1}]$.
- d_2 —the number of j such that $(c_j - b_j)/c_j \in (\frac{2n-1}{3n-1}, 1]$.

We have

$$\begin{aligned} d_0 + d_1 + d_2 &= n \\ \frac{n-1}{3n-1}d_1 + \frac{2n-1}{3n-1}d_2 &< \sum_{j=1}^N (c_j - b_j)/c_j \leq \sum_{j=1}^N (c_j - b_j) \quad \text{since } c_j \geq 1 \\ &\leq \frac{2n}{3n-1}k \quad \text{by assumption} \\ &\Rightarrow (1/2 - 1/2n)d_1 + (1 - 1/2n)d_2 < k \\ &\Rightarrow (d_0 + d_1 + d_2) - (1/2 - 1/2n)d_1 - (1 - 1/2n)d_2 > n - k \\ &\Rightarrow d_1/2 + d_0 + (d_1 + d_2)/2n > n - k \end{aligned}$$

Since $d_1 + d_2 \leq n$, this implies

$$d_1/2 + d_0 + 1/2 > n - k$$

If d_1 is even, then this implies

$$d_1/2 + d_0 \geq n - k$$

If d_1 is odd, then the left-hand side is integer, and we get

$$d_1/2 + d_0 + 1/2 \geq n - k + 1 \Rightarrow (d_1 - 1)/2 + d_0 \geq n - k.$$

In both cases

$$\lfloor d_1/2 \rfloor + d_0 \geq n - k.$$

There are d_0 indices j for which $b_j/c_j \geq 1 - \frac{n-1}{3n-1} = \frac{2n}{3n-1}$, so $b_j \geq \frac{2n}{3n-1}$ too.

There are also d_1 indices j for which $b_j/c_j \geq 1 - \frac{2n-1}{3n-1} = \frac{n}{3n-1}$, so $b_j \geq \frac{n}{3n-1}$ too. Pairing these elements gives $\lfloor d_1/2 \rfloor$ pairs with a sum of at least $\frac{2n}{3n-1}$. All in all, there are $\lfloor d_1/2 \rfloor + d_0 \geq n - k$ sets (singletons or pairs) with sum at least $\frac{2n}{3n-1}$. \square

Corollary A.6. *The Lone Divider algorithm can attain a fair allocation of discrete goods with $t_i = \frac{2n}{3n-1} \text{MMS}_i^{1\text{-out-of-}n}(C) > \frac{2}{3} \text{MMS}_i^{1\text{-out-of-}n}(C)$.*

Proof. By Theorem 4.1, it is sufficient to prove that the t_i are reasonable thresholds for each agent i . Let $(C_j)_{j \in [n]}$ be a 1-out-of- n MMS partition of i . Let $c_j := V_i(C_j) / \text{MMS}_i^{1\text{-out-of-}n}(C)$. By definition of the MMS, $c_j \geq 1$ for all $j \in [n]$.

For every $k \geq 0$, suppose we remove some objects whose total value is at most $\frac{2n}{3n-1}k \cdot \text{MMS}_i^{1\text{-out-of-}n}(C)$. Let $b_j :=$ the total value remaining in C_j , divided by $\text{MMS}_i^{1\text{-out-of-}n}(C)$; so $b_j \in [0, c_j]$, and the sum of b_j is at least $\sum_j c_j - \frac{2n}{3n-1}k$. Lemma A.5 implies that the b_j can be partitioned into $n - k$ subsets with a sum of at least $\frac{2n}{3n-1}$. This corresponds to a partitioning of the remaining objects into $n - k$ bundles with a value of at least t_i . Therefore, both conditions (1) and (2) in Definition 4.1 are satisfied. \square

Appendix B. Related concepts

B.1. Similar concepts with a different name

Concepts similar to envy-free matching appeared in previous papers related to fair division, but they were “hidden” inside proofs of more specific algorithms. One goal of the present paper is to uncover these hidden gems.

As far as we know, the earliest concept similar to envy-free matching was presented by Kuhn [27] [unnumbered lemma in page 31]. Kuhn presents the lemma in matrix form. 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 Corollary 1.1(b). Kuhn used this lemma in an algorithm for fair cake-cutting, which is now known as the Lone Divider algorithm (see Section 4).

Procaccia and Wang [32] [sub.3.1] mention another concept similar to envy-free matching, among proofs of other lemmas related to fair allocation of discrete objects. They constructed a particular bipartite graph that admits a perfect matching between a subset $X^* \subseteq X$ and a subset $Y^* \subseteq Y$, where there are no edges between $X \setminus X^*$ and Y^* ; their X^* and Y^* correspond to our X_L and Y_L , and the “no edges” property corresponds to our Theorem 1.1(a). Since they were mainly interested in the case of a constant number of players, for which $|X|$ is constant, they did not consider efficient algorithms for computing such a matching.¹²

Later, Amanatidis et al. [2] [lem.4.5] improved this construction and presented a polynomial-time algorithm for finding a non-empty envy-free matching. Their $X \setminus X^+$ and $Y \setminus \Gamma(X^+)$ correspond to our X_L and Y_L respectively, and their APX-MMS algorithm corresponds to the Lone Divider algorithm (Algorithm 4) with threshold values corresponding to $2/3$ fraction of the 1-out-of- n MMS (see Appendix A).

Later, Ghodsi et al. [18] [def.3.5] presented a construction that corresponds to our construction as follows: given a maximum-cardinality matching M , $F_H(M, \hat{X})$ corresponds to Y_L ; $N(F_H(M, \hat{X}))$ corresponds to X_L ; ¹³ $\hat{Y}_1 \cup \hat{Y}_2$ corresponds to X_S ; and $\hat{X} \setminus F_H(M, \hat{X})$ corresponds to Y_S . Their Lemmas 3.6, 3.7, 3.8 correspond to Theorem 1.1 parts (b), (c), (e).

Recently, Bogomolnaia et al. [9] presented a similar concept which they called a “proper matching”, for finding a min-max cake allocation when agents have general valuations.

Since all these authors used envy-free matching mainly as an intermediate step in a larger algorithm, they did not consider questions such as the uniqueness of the partition, and did not attempt to find an envy-free matching of maximum cardinality or minimum cost.

Our presentation of envy-free matching as a stand-alone graph-theoretic concept allows us to both simplify old algorithms and design new ones, as illustrated in Sections 4 and 5.

B.2. Different concepts with a similar name

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

(1) 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-

¹² This construction did not appear in the journal version [28].

¹³ In fact, as we prove in Theorem 1.1, Y_L and X_L depend only on G and are independent of M .

prices (and leave some items with positive price unallocated) in order to increase his expected revenue. See, for example, Guruswami et al. [19], Alaei et al. [1].

(2) 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 [40,41]. 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.

(3) In contrast, our envy-free matching is an abstract graph-theoretic concept: it is defined for any bipartite graph, and does not require any notion of a price or a ranking.

To differentiate the terms, one can use, for example:

For (1)—“price envy-free matching” or “market envy-free matching”;

For (2)—“no-justified-envy matching” or “justified-envy-free matching”;

For (3)—“binary envy-free matching” or “unweighted envy-free matching”.

References

- [1] S. Alaei, K. Jain, A. Malekian, Competitive Equilibria in Two Sided Matching Markets with Non-transferable Utilities, arXiv preprint 1006.4696, 2012.
- [2] G. Amanatidis, E. Markakis, A. Nikzad, A. Saberi, Approximation algorithms for computing maximin share allocations, *ACM Trans. Algorithms* 13 (4) (2017) 52.
- [3] S. Avvakumov, R. Karasev, Equipartition of a segment, arXiv preprint arXiv:2009.09862.
- [4] H. Aziz, S. Gaspers, S. Mackenzie, T. Walsh, Fair assignment of indivisible objects under ordinal preferences, *Artif. Intell.* 227 (2015) 71–92, ISSN 00043702, arXiv preprint 1312.6546.
- [5] M. Babaioff, N. Nisan, I. Talgam-Cohen, Competitive Equilibria with Indivisible Goods and Generic Budgets, arXiv preprint 1703.08150 v1 (2017–03), 2017.
- [6] M. Babaioff, N. Nisan, I. Talgam-Cohen, Fair Allocation through Competitive Equilibrium from Generic Incomes, in: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, ACM, 180–180, arXiv preprint 1703.08150 v2 (2018–09), 2019.
- [7] S. Barman, S.K. Krishnamurthy, Approximation algorithms for maximin fair division, in: *Proceedings of the 2017 ACM Conference on Economics and Computation*, ACM, 2017, pp. 647–664.
- [8] A. Beynier, Y. Chevaleyre, L. Gourvès, A. Harutyunyan, J. Lesca, N. Maudet, A. Wilczynski, Local envy-freeness in house allocation problems, *Auton. Agent. Multi-Agent Syst.* 33 (5) (2019) 591–627.
- [9] A. Bogomolnaia, H. Moulin, R. Stong, Guarantees in Fair Division: general or monotone preferences, arXiv preprint arXiv:1911.10009.
- [10] S.J. Brams, A.D. Taylor, *Fair Division: From Cake Cutting to Dispute Resolution*, Cambridge University Press, Cambridge UK, ISBN 0521556449, 1996.
- [11] E. Budish, The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes, *Journal of Political Economy* 119 (6) (2011) 1061–1103, ISSN 00223808.
- [12] G. Chèze, Don't cry to be the first! Symmetric fair division algorithms exist, arXiv preprint 1804.03833, 2018.
- [13] J.P. Dickerson, J. Goldman, J. Karp, A.D. Procaccia, T. Sandholm, The Computational Rise and Fall of Fairness, in: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, AAAI Press, 1405–1411, 2014.
- [14] E. Elkind, E. Segal-Halevi, W. Suksompong, Graphical Cake Cutting via Maximin Share, in: *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI '21)*, arXiv preprint 2105.04755, 2021.
- [15] S. Even, A. Paz, A Note on Cake Cutting, *Discrete Appl. Math.* 7(3) (1984) 285–296, ISSN 0166218X.
- [16] J. Gan, W. Suksompong, A.A. Voudouris, Envy-Freeness in House Allocation Problems, *Math. Soc. Sci.* 101 (2019) 104–106.
- [17] J. Garg, P. McGlaughlin, S. Taki, Approximating maximin share allocations, Open access series in informatics 69.
- [18] M. Ghodsi, M. HajiAghayi, M. Seddighin, S. Seddighin, H. Yami, Fair Allocation of Indivisible Goods: Improvements and Generalizations, in: *Proceedings of the 2018 ACM Conference on Economics and Computation*, ACM, 539–556, arXiv preprint 1704.00222, 2018.
- [19] V. Guruswami, J.D. Hartline, A.R. Karlin, D. Kempe, C. Kenyon, F. McSherry, On Profit-maximizing Envy-free Pricing, in: *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1164–1173, ISBN 0-89871-585-7, 2005.
- [20] P. Hall, On representatives of subsets, in: *Classic Papers in Combinatorics*, Springer, 58–62, in collection from 2009, 1935.
- [21] J.E. Hopcroft, R.M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* 2 (1973) 225–231, ISSN 0097-5397, 10.1137/0202019.
- [22] H. Hosseini, A. Searns, E. Segal-Halevi, Ordinal Maximin Share Approximation for Goods, arXiv preprint arXiv:2109.01925.
- [23] H. Hosseini, S. Sikdar, R. Vaish, J. Wang, L. Xia, Fair Division through Information Withholding, arXiv preprint 1907.02583, 2019.
- [24] X. Huang, P. Lu, An algorithmic framework for approximating maximin share allocation of chores, arXiv preprint arXiv:1907.04505.
- [25] R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, K.E. Paluch, Rank-maximal matchings, *ACM Trans. Algorithms* 2 (4) (2006) 602–610.
- [26] N. Kamiyama, P. Manurangsi, W. Suksompong, On the complexity of fair house allocation, *Oper. Res. Lett.*
- [27] H.W. Kuhn, On games of fair division, Princeton University Press, 29–37, 1967.
- [28] D. Kurokawa, A.D. Procaccia, J. Wang, Fair Enough: Guaranteeing Approximate Maximin Shares, *J. ACM* 65 (2) (2018) 8.
- [29] Z. Luria, Envy-free matching, Mathematics Stack Exchange, URL: <https://math.stackexchange.com/q/621306/29780> (version: 2013-12-29), 2013.
- [30] Y. Manabe, T. Okamoto, Meta-Envy-Free Cake-Cutting Protocols, in: P. Hliněný, A. Kučera (Eds.), *Mathematical Foundations of Computer Science 2010*, vol. 6281 of Lecture Notes in Computer Science, chap. 44, Springer, Berlin Heidelberg, Berlin, Heidelberg, 501–512, ISBN 978-3-642-15154-5, 2010.
- [31] P. Manurangsi, W. Suksompong, When do envy-free allocations exist?, *Proceedings of the AAAI Conference on Artif. Intell.* 33 (2019) 2109–2116.
- [32] A.D. Procaccia, J. Wang, Fair Enough: Guaranteeing Approximate Maximin Shares, in: *Proc. EC-14*, ACM, New York, NY, USA, 675–692, ISBN 978-1-4503-2565-3, 2014.
- [33] W.R. Pulleyblank, Matchings and extensions, *Handb. Combinatorics* 1 (1995) 179–232.
- [34] L. Ramshaw, R. Tarjan, On Minimum-Cost Assignments in Unbalanced Bipartite Graphs, technical report, 2012.
- [35] J.M. Robertson, W.A. Webb, *Cake-Cutting Algorithms: Be Fair if You Can*, A K Peters/CRC Press, first ed., ISBN 1568810768, 1998.
- [36] E.L. Schreiber, R.E. Korf, M.D. Moffitt, Optimal multi-way number partitioning, *J. ACM* 65 (4) (2018) 1–61.

- [37] E. Segal-Halevi, Fair Multi-Cake Cutting, *Discrete Applied Mathematics* 291 (2021) 15–35, arXiv preprint 1812.08150.
- [38] H. Steinhaus, [The problem of fair division](#), *Econometrica* 16 (1) (1948) 101–104.
- [39] G.J. Woeginger, A polynomial-time approximation scheme for maximizing the minimum machine completion time, *Oper. Res. Lett.* 20(4) (1997) 149–154, ISSN 01676377.
- [40] Q. Wu, A.E. Roth, The lattice of envy-free matchings, *Games Econ. Behav.* 109 (2018) 201–211.
- [41] Y. Yokoi, [Envy-free matchings with lower quotas](#), *Algorithmica* 82 (2) (2020) 188–211.