Maxileximin Envy Allocations and Connected Goods

Gianluigi Greco, Francesco Scarcello

University of Calabria

{gianluigi.greco, francesco.scarcello}@unical.it

Abstract

Fair allocation of indivisible goods presents intriguing challenges from both a social choice perspective and an algorithmic standpoint. Due to the indivisibility of goods, it is common for one agent to envy the bundle of goods assigned to another agent and, indeed, envy-free solutions do not exist in general. In line with the classical game-theoretic concept of Nucleolus in coalitional games, we propose that a fair allocation should minimize the agents' dissatisfaction profile in a lexicographic manner, where the dissatisfaction of an agent is defined as her maximum envy towards other agents. Therefore, we seek allocations that minimize the maximum envy. In cases where multiple solutions have an equal maximum value, we minimize the second-worst value, and so on. Additionally, as is customary in fair division problems, we also consider an efficiency requirement: among the allocations with the best agents' dissatisfaction profile, we prioritize those that maximize the sum of agents' utilities, known as maximum social welfare. Such allocations, referred to as *maxileximin* allocations, always exist.

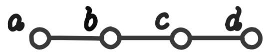
In this study, we analyze the computational properties of maxileximin allocations in the context of fair allocation problems with constraints. Specifically, we focus on the *Connected Fair Division* problem, where goods correspond to the nodes of a graph, and a bundle of goods is allowed if the subgraph formed by those goods is connected. We demonstrate that the problem is $\mathrm{F}\Delta_2^{\mathrm{P}}$-complete, even for instances with simple graphical structures such as path and star graphs. However, we identify islands of tractability for instances with more intricate graphs, such as those having bounded treewidth, provided that the number of agents is bounded by a fixed number and utility functions use small values.

Introduction

Fair allocation of indivisible goods is a central problem in social choice theory (Amanatidis et al. 2023; Steinhaus 1948; Brams and Taylor 1996), and it has been intensively analysed from the computational and algorithmic viewpoint by the AI community (Bouveret et al. 2016; Lang and Rothe 2016; Walsh 2020; Shams et al. 2022). With multiple agents and indivisible goods, we cannot expect to be able to allocate goods in a way that makes everyone happy, so that in general some agent could envy the bundle of goods assigned to another agent. However, if we can find an *envy-free* (EF) allocation where no agent strictly prefers the bundle assigned of another agent to her own bundle, such an allocation can be clearly perceived as a *fair* one. Whenever these allocations do not exist, we have to find some other fair way for assigning goods. In fact, there are many interesting works in the literature which define relaxations of envy-freeness (cf. Amanatidis, Birmpas, and Markakis 2018), notions such as envy-freeness up to one good (Lipton et al. 2004) or up to any good (Caragiannis et al. 2016), or the related maximin share allocations (Budish 2011). Other approaches deal explicitly with the (unavoidable) envy, by trying to guarantee to agents the maximum amount of fairness with respect to the scenario at hands. For instance, one can minimize the maximum envy ratio (Caragiannis et al. 2009), the multiplicative degree of envy (Nguyen and Rothe 2014), the number of envious agents (Netzer, Meisels, and Zivan 2016), or the maximum envy (e.g., Cai, Filos-Ratsikas, and Tang 2016), just to name a few.

This work pushes ahead on this research and aims at identifying a solution that exists always and is as fair and acceptable as possible for all agents. Fairness properties are well studied in coalitional game theory and fundamental to solution concepts, such as the classical notion of *Nucleolus* (Schmeidler 1969), which minimizes in a lexicographic way the dissatisfaction of all player coalitions. In our context, the dissatisfaction can be measured as the maximum envy towards other agents. Following this approach, we argue that an important class of allocations, which received less attention in the literature, is the one of the *leximin* allocations, that is, those minimizing the maximum envy and, among possible solutions with equal maximum value, minimize the second-worst value, and so on.

Example 1. Consider a scenario where four indivisible goods (say a, b, c, and d) are available. Figure 1 reports on the left the utilities that three agents can get with these goods. E.g., agent 1 values 3 the good a, and 2 the other goods. Clearly, in any feasible allocation, one agent gets a bundle with two goods, while the others get one good. The figure also displays two allocations $\mathbf{B}$ and $\mathbf{B}'$. According to the former, agent 1 and agent 3 have utility 3, while agent 2 gets the bundle $\{b, c\}$ with a total utility 4. Both agents 1 and 3 would get utility 4 with the bundle assigned to agent 2, hence the difference $4-3$ is a measure of their envy. Agent 2



	a	b	c	d
1	$\boxed{3}$	2	2	2
2	2	$\boxed{2}$	$\boxed{2}$	2
3	2	2	2	$\boxed{3}$

	a	b	c	d
1	$\boxed{3}$	$\boxed{2}$	2	2
2	2	2	$\boxed{2}$	2
3	2	2	2	$\boxed{3}$

Figure 1: Allocations $\mathbf{B}$ and $\mathbf{B}'$ in Example 1

does not envy any bundle of other agents. Consider now $\mathbf{B}'$: this is good division for agent 1, who get the bundle $\{a, b\}$ with a total utility 5. Agent 2 envies that bundle, and her envy is $4 - 2 = 2$ in this case. The envy of agent 3 is instead $4 - 3 = 1$. Therefore it is natural to prefer the allocation $\mathbf{B}$ to $\mathbf{B}'$. On the other hand, $\mathbf{B}'$ should be preferred to the allocation $\mathbf{B}''$, symmetrical to $\mathbf{B}$, where agent 1 takes d and agent 3 takes a. In this case, we would have two unhappy agents (1 and 3) with the worst envy level 2, instead of 1. $\triangleleft$

While focusing on the kinds of setting exemplified above, it must be noticed that—unlike the Nucleolus that is defined in a fully transferable utility setting and it is a unique point—multiple lexmin allocations can exist over the same scenario. In particular, minimizing the envy does not always lead to maximize the sum of agents' utilities, that is, the *social welfare*. We thus explicitly require that the desirable allocations, called MAXILEXIMIN allocations (short: MLM), besides having the lexicographically smallest envy vector, must have the maximum social welfare, too. Indeed, such efficiency requirement is typically needed in desirable outcomes, otherwise there could be solutions without envies just because all agents are unhappy. In fact, fairness is usually combined with efficiency, see (Barman, Krishnamurthy, and Vaish 2018; Caragiannis et al. 2016; de Keijzer et al. 2009; Bei et al. 2021).

In this paper, we precisely focus on MLM allocations and we consider the setting of fair allocation problems with constraints, which received considerable attention in the last few years (see, e.g., Suksompong 2021). In particular, we deal with the Connected Fair Division (CFD) problem proposed by Bouveret et al. (2017), where goods correspond to the nodes of a *graph*, and a bundle of goods is allowed if the subgraph induced by its goods is connected. For instance, in Figure 1 the four goods are arranged on a path, so that $\{a, b\}$ is a feasible bundle, while $\{a, c\}$ is not. This setting attracted the attention of the community, and there are many works in the literature, studying relaxation of EF allocations such as the maximin share, the parameterized complexity by considering different possible fixed parameters, the price of connectivity, and so on (Suksompong 2017; Lonc and Truszczynski 2018; Bouveret, Cechlárová, and Lesca 2018; Igarashi and Peters 2019; Bilò et al. 2022; Deligkas et al. 2021; Bei et al. 2022).

In fact, an approach to envy minimization similar to ours has recently been defined in (Shams et al. 2021), following the older social-evaluation function based on Gini inequality indices (Weymark 1981): the work aims at reducing the vector of user dissatisfactions by minimizing the *Ordered Weighted Average* (OWA) of the envy vector, that is, by using an aggregation function that computes the weighted sum of the envy vector of all agents, non-increasingly ordered. The use of decreasing weight vectors (that sum to one) allows obtaining solutions where the dissatisfaction reduction is preferred more for those who are unhappier. It is shown that allocations minimizing such an OWA can be computed using a mixed-integer linear program. The work of Shams et al. (2021), however, does not consider fair division with constraints; in particular, as far as we know, connectivity constraints have not been studied for OWA allocations.

In the following, we refer to MLM-CFD as the problem of computing a maxileximin allocation in the setting of connected fair division. We study MLM-CFD from the computational viewpoint and we provide the following contributions:

- We show that MLM-CFD is an intractable problem, precisely $\mathrm{F}\Delta_2^{\mathrm{P}}$-complete, even on path and star graphs (since an MLM always exists, the decision problem is trivial). These results do not rely on the known NP-hardness of deciding whether envy-free allocations on such graphs exist (Bouveret et al. 2017), since $\mathrm{F}\Delta_2^{\mathrm{P}}$-hardness is shown to hold even on settings where envy-free allocations are guaranteed to exist, and the problem is just maximizing the social welfare over them (which are clearly the leximin ones).
- We then consider possible restrictions of the problem, but it turns out that $\mathrm{F}\Delta_2^{\mathrm{P}}$-hardness still holds even on *smooth scenarios* where utility functions use only "small" values, i.e., values that are polynomially bounded (or, equivalently, given in unary notation). This time, however, more complex network topologies are involved.
- Finally, we analyze scenarios where the number of agents is bounded by some given constant to identify a tractable class of MLM-CFD instances. We prove that MLM-CFD belongs to the functional version of LogCFL, a complexity class included in polynomial time and whose problems can be solved by leveraging parallelization (Gottlob, Leone, and Scarcello 2002; Chandra, Kozen, and Stockmeyer 1981), if we consider smooth scenarios and the graphs of the goods have small degree of cyclicity, formally, if they have bounded *treewidth* (Robertson and Seymour 1986). Interestingly, the algorithm can be adapted to work for different fairness notions, in particular for OWA allocations; moreover, smooth functions clearly include binary evaluation functions, which are also studied in the literature (see, e.g., Goldberg, Hollender, and Suksompong 2020). The problem is still intractable, if utility functions use arbitrary large values.

Formal Framework

Envy-free allocations. An *allocation scenario* for MLM-CFD is a tuple $\sigma = (N, G, \{u_i\}_{i \in N})$ where N is the set $[1; n]$ of natural numbers encoding the agents, $G = (V, E)$ is a connected non-empty graph whose nodes in V are the available goods and, for each $i \in N$, u_i is the *utility function* of agent i. In this paper, we focus on the widely used framework where the value of a bundle of goods for any agent $i \in N$ is obtained as the sum of the values she assigns to each good: u_i is a function that maps each good $v \in V$ to a natural[1] number $u_i(v)$, and the value $u_i(X)$ of a set X of goods is just given by $\sum_{g\in X} u_i(g)$, with $u_i(\emptyset) = 0$. An *allocation* for σ is a tuple $\mathbf{B} = (\mathbf{B}_1, ..., \mathbf{B}_n)$, whose components associate each agent $i \in N$ with a *bundle* $\mathbf{B}_i \subseteq V$. In the framework we consider, bundles are required to be connected, that is, the goods in each bundle induce a connected subgraph of G. Since goods are indivisible, bundles must be disjoint, that is, $\mathbf{B}_i \cap \mathbf{B}_{i'} = \emptyset$ for each pair of distinct agents i and i'. An allocation $\mathbf{B}$ is *envy-free* if $u_i(\mathbf{B}_i) \geq u_i(\mathbf{B}_j)$, for each pair $i, j \in N$. As it is usually done when dealing with envy-freeness, we also require that all goods are assigned to agents, so that allocations are *complete*: $\bigcup_{i=1}^n \mathbf{B}_i = G$.

Example 2. Consider again the example in Figure 1: allocations $\mathbf{B}$ and $\mathbf{B}'$ are complete, and we have $\mathbf{B}_1 = \{a\}$, $\mathbf{B}_2 = \{b, c\}$, $\mathbf{B}_3 = \{d\}$, with $u_1(\mathbf{B}_1) = 3$, $u_2(\mathbf{B}_2) = 4$, and $u_3(\mathbf{B}_3) = 3$. Note that neither allocation is envy-free.

Lexicographic Envy Minimization. The *envy* of agent i for an allocation $\mathbf{B}$ is the non-negative value $\max_{j\in N} u_i(\mathbf{B}_j) - u_i(\mathbf{B}_i)$. Define $\xi(\mathbf{B})$ to be the vector containing the agents' envies for $\mathbf{B}$ arranged in non-increasing order. For a pair of n-dimensional vectors v_1, v_2, denote by $v_1 \prec v_2$ the fact that v_1 precedes v_2 in the (total) lexicographic order, that is, there exists some $q \in N$ such that $v_1[p] = v_2[p]$ for all $p < q$, and $v_1[q] < v_2[q]$. An allocation $\mathbf{B}$ is said a *leximin* allocation if there is no allocation $\mathbf{B}'$ such that $\xi(\mathbf{B}') \prec \xi(\mathbf{B})$. The set of all leximin allocations is denoted by $\text{LEXIMIN}(\sigma)$.

Example 3. For the considered allocations, we have $\xi(\mathbf{B}) = \langle 1, 1, 0\rangle$, $\xi(\mathbf{B}') = \langle 2, 1, 0\rangle$, and $\xi(\mathbf{B}'') = \langle 2, 2, 0\rangle$. Therefore, $\xi(\mathbf{B}) \prec \xi(\mathbf{B}') \prec \xi(\mathbf{B}'')$. It can be checked that $\mathbf{B}$ is the unique leximin allocation for the scenario. $\lhd$

Refinements based on the Social Welfare. The *social welfare* of an allocation $\mathbf{B}$ is the value $\text{SW}(\mathbf{B}) = \sum_{i=1}^n u_i(\mathbf{B}_i)$. E.g., in our running example, we have $\text{SW}(\mathbf{B}) = 3+4+3 = 10$. A leximin allocation $\mathbf{B}$ is SW-maximal, or *maxileximin*, if $\text{SW}(\mathbf{B}) \geq \text{SW}(\mathbf{B}')$, for every $\mathbf{B}' \in \text{LEXIMIN}(\sigma)$. The set of these allocations is denoted by $\text{MAXILEXIMIN}(\sigma)$, in the running example it is the singleton $\{\mathbf{B}\}$.

Arbitrary Number of Agents

We assume a standard encoding for any allocation scenario σ: the utility function u_i of each agent $i \in N$ is encoded by explicitly listing all possible goods v with the associated value $u_i(v)$. Therefore, the encoding size $||\sigma||$ is $O(|V|^2 + |N| \times |V| \times M_\sigma)$, where M_σ is the maximum size over the values encodings occurring in the utility functions.

Arbitrary Utility Functions

Our first result is that MLM-CFD is complete for $\text{F}\Delta_2^{\text{P}}$, the class of all problems for which a solution can be computed in polynomial time by invoking with unitary cost an NP oracle. Notably, hardness holds even on the restriction of MLM-CFD to instances where envy-free allocations are guaranteed to exist (shortly denoted by MLM-CFD$_{ef}$), and even on very simple graph topologies—namely star and path graphs.

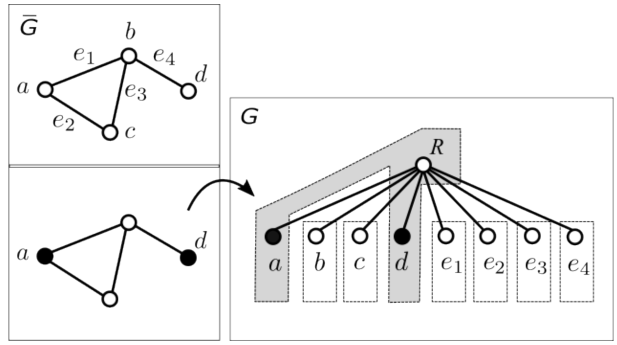


Figure 2: Construction in the proof of Theorem 4.

Theorem 4. *MLM-CFD$_{ef}$ is $\text{F}\Delta_2^{\text{P}}$-hard on star graphs.*

Proof Sketch. Let $\bar{G} = (\bar{V}, \bar{E})$ be a graph where each node $x \in \bar{V}$ is weighted with a number $w_x \geq 0$. A set of nodes $\bar{V}' \subseteq \bar{V}$ is an *independent set* if $|\bar{V}' \cap e_i| \leq 1$, for each $e_i \in \bar{E}$. The *weight* of an independent set $\bar{V}' \subseteq \bar{V}$ is the value $\sum_{x\in\bar{V}'} w_x$. Computing an independent set having maximum possible weight is $\text{F}\Delta_2^{\text{P}}$-hard (cf., Krentel 1988).

Based on $\bar{G}$, we build (in polynomial time) an allocation scenario $\sigma_{\bar{G}} = (N, G, \{u_i\}_{i\in N})$ as follows. The graph $G = (V, E)$ is a star, having the node $R \in V$ as its center, and such that $V = \{R\} \cup \bar{E} \cup \bar{V}$. Note that nodes and edges of $\bar{G}$ are viewed as goods in the new instance (nodes of G). As an example, Figure 2 reports a graph $\bar{G}$ over nodes $\{a, b, c, d\}$ and the graph associated with the corresponding scenario $\sigma_{\bar{G}}$. The set N of agents in $\sigma_{\bar{G}}$ is such that:

- for each $e_i = \{x, y\} \in \bar{E}$, N contains the agent $\text{AG}(e_i)$ such that $u_{\text{AG}(e_i)}(x) = u_{\text{AG}(e_i)}(y) = 2$, $u_{\text{AG}(e_i)}(e_i) = 3$, and $u_{\text{AG}(e_i)}(v) = 0$ for each $v \in V \setminus \{x, y, e_i\}$;
- N contains the agent $\text{AG}(R)$ such that: $u_{\text{AG}(R)}(R) = \sum_{x\in\bar{V}} w_x + 1$; $u_{\text{AG}(R)}(x) = w_x$ for each $x \in \bar{V}$; and $u_{\text{AG}(R)}(v) = 0$ for each $v \in V \setminus \bar{V} \setminus \{R\}$;
- N contains further dummy agents $\text{AG}(1), ..., \text{AG}(|\bar{V}|-1)$ getting utility 0 for any good;
- no further agent is in N.

Let us point out two important properties of the reduction.

Assume that $\bar{V}'$ is an independent set for the graph $\bar{G}$ and consider an allocation $\mathbf{B}$ such that: $\mathbf{B}_{\text{AG}(e_i)} = \{e_i\}$ for each $e_i \in \bar{E}$; $\mathbf{B}_{\text{AG}(R)} = \{R\} \cup \bar{V}'$; and the remaining $|\bar{V}| - |\bar{V}'|$ goods are arbitrarily allocated to agents $\text{AG}(1), ..., \text{AG}(|\bar{V}| - |\bar{V}'|)$. It is immediate to check that $\mathbf{B}$ is envy-free. So, any solution to MLM-CFD is an envy-free allocation maximizing the social welfare. In particular, the social welfare of $\mathbf{B}$ is given by $\text{SW}(\mathbf{B}) = \sum_{x\in\bar{V}'} w_x + 3|\bar{E}| + u_{\text{AG}(R)}(R)$.

On the other hand, let $\mathbf{B}$ be an envy-free allocation and note that $R \in \mathbf{B}_{\text{AG}(R)}$. Since bundles are connected, for each $e_i = \{x, y\} \in \bar{E}$, agent $\text{AG}(e_i)$ cannot simultaneously

[1]For the sake of presentation, to avoid discussing subtle issues related to the fractional form representation of rational numbers, we prefer (w.l.o.g.) to use natural numbers encoded in binary.

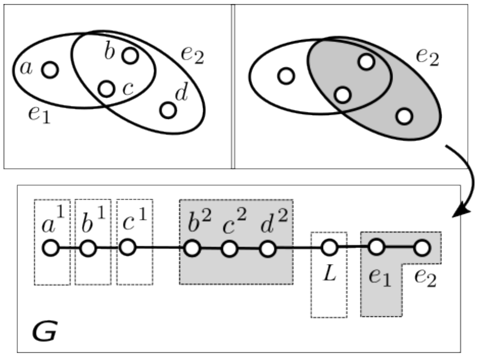


Figure 3: Construction in the proof of Theorem 5.

get x and y. Hence, we derive that $\mathbf{B}_{\text{AG}(e_i)} = \{e_i\}$. Moreover, since $\text{AG}(e_i)$ does not envy $\text{AG}(R)$, it is the case that $|\mathbf{B}_{\text{AG}(R)} \cap e_i| \leq 1$. That is, $\mathbf{B}_{\text{AG}(R)} \cap \bar{V}$ is an independent set for the graph $\bar{G}$. And, indeed, the weight of this independent set is given by $\text{SW}(\mathbf{B}) - 3|\bar{E}| - u_{\text{AG}(R)}(R)$.

Hence, from any solution to MLM-CFD on $\sigma_{\bar{G}}$, we get an independent set for $\bar{G}$ having maximum weight. $\square$

Theorem 5. MLM-CFD$_{ef}$ *is* $\text{F}\Delta_2^{\text{P}}$*-hard on path graphs.*

Proof Sketch. Let $\bar{H} = (\bar{V}, \bar{E})$ be a hypergraph where $\bar{E} = \{e_1, ..., e_m\}$ is a set of hyperedges such that, for each $e_i \in \bar{E}$, $|e_i \cap V| = 3$. Assume that each hyperedge $e_i \in \bar{E}$ is equipped with a weight $w_i \geq 0$. A *set packing* for $\bar{H}$ is a set $E' \subseteq \bar{E}$ of hyperedges that are pairwise disjoint, and its *weight* is the value $\sum_{e_i \in E'} w_i$. Computing a set packing having maximum weight is $\text{F}\Delta_2^{\text{P}}$-hard (cf., Krentel 1988).

Based on $\bar{H}$, we build (in polynomial time) an allocation scenario $\sigma_{\bar{H}} = (N, G, \{u_i\}_{i \in N})$ as follows. The graph $G = (V, E)$ is a path over $V = \{x^i, y^i, z^i, e_i \mid e_i = \{x, y, z\} \in \bar{E}\} \cup \{L\}$. In particular, the subgraph induced over x^i, y^i, z^i is connected, for each $e_i = \{x, y, z\} \in \bar{E}$; and L separates the nodes in $\{e_1, ..., e_m\}$ from the others—the specific ordering of the nodes in the path is irrelevant as long as these properties are satisfied. As an example, Figure 3 reports a hypergraph $\bar{H}$ over nodes $\{a, b, c, d\}$ and the graph associated with the corresponding scenario $\sigma_{\bar{H}}$.

The set N of agents in $\sigma_{\bar{H}}$ is such that:

- for each $e_i \in \bar{E}$, N contains $\text{AG}(e_i)$ with $u_{\text{AG}(e_i)}(x^i) = u_{\text{AG}(e_i)}(y^i) = u_{\text{AG}(e_i)}(z^i) = 2w_i$, $u_{\text{AG}(e_i)}(e_i) = 5w_i$; and $u_{\text{AG}(e_i)}(v) = 0$ for each $v \notin \{x^i, y^i, z^i, e_i\}$;
- for each $x \in \bar{V}$ occurring in δ hyperedges, N contains the agents $\text{AG}(x, 1), ..., \text{AG}(x, \delta - 1)$. Each $\text{AG}(x, j)$ gets utility 0 for any good, but $u_{\text{AG}(x,j)}(v) = 1$ for $v \in \{x^h \mid x \in e_h, e_h \in \bar{E}\}$;
- N contains agent $\text{AG}(L)$ that gets utility 0 for any good, but $u_{\text{AG}(L)}(L) = 1$;
- no other agent is in N.

Let us point out two important properties of the reduction. Assume that $\bar{E}'$ is a set packing of maximum weight and note that an allocation $\mathbf{B}$ satisfying the following conditions can be built given $\bar{E}'$: for each $e_i = \{x, y, z\} \in \bar{E}'$, $\mathbf{B}_{\text{AG}(e_i)} \supseteq \{x^i, y^i, z^i\}$; for each $e_i = \{x, y, z\} \notin \bar{E}'$, $\mathbf{B}_{\text{AG}(e_i)} \supseteq \{e_i\}$; $\mathbf{B}_{\text{AG}(L)} \supseteq \{L\}$; for each $x \in \bar{V}$ occurring in $\delta > 1$ hyperedges, $\mathbf{B}_{\text{AG}(x,1)} \cup \ldots \cup \mathbf{B}_{\text{AG}(x,\delta-1)} \supseteq \{x^h \mid x \in e_h, e_h \in \bar{E} \setminus \bar{E}'\}$. It can be checked that these conditions guarantee that $\mathbf{B}$ is envy-free. In particular, note that if $e_i = \{x, y, z\}$ is not in $\bar{E}'$, then no agent can get a bundle with the three goods x^i, y^i, and z^i. By simple algebraic calculations, it can be checked that: $\text{SW}(\mathbf{B}) = 6\sum_{e_i \in \bar{E}'} w_i + 5\sum_{e_i \notin \bar{E}'} w_i + 1 + 3|\bar{E}| - |\bar{V}| = \sum_{e_i \in \bar{E}'} w_i + \sum_{e_i \in \bar{E}} w_i + 1 + 3|\bar{E}| - |\bar{V}|$.

On the other hand, assume that $\mathbf{B}$ is an envy-free allocation. We immediately derive that $\mathbf{B}_{\text{AG}(L)} \supseteq \{L\}$, and $|(\mathbf{B}_{\text{AG}(x,1)} \cup \ldots \cup \mathbf{B}_{\text{AG}(x,\delta-1)}) \cap \{x^h \mid x \in e_h, e_h \in \bar{E}\}| = \delta - 1$ for each $x \in \bar{V}$ occurring in $\delta > 1$ hyperedges. Let S be the set of goods not allocated to any of these agents. Assume that $\mathbf{B}$ maximizes the social welfare. If S contains the nodes x^i, y^i, and z^i, then $\mathbf{B}_{\text{AG}(e_i)} \supseteq \{x^i, y^i, z^i\}$ clearly holds. Otherwise, we have $\mathbf{B}_{\text{AG}(e_i)} \supseteq \{e_i\}$. In particular, note that $|\{x^h \mid x \in e_h, e_h \in \bar{E}\} \cap S| \leq 1$ holds for each $x \in \bar{V}$ occurring in $\delta > 1$ hyperedges. Therefore, the set $\{e_i \mid \mathbf{B}_{\text{AG}(e_i)} \supseteq \{x^i, y^i, z^i\}\}$ is a set packing. It can be checked that its weight is $\text{SW}(\mathbf{B}) - \sum_{e_i \in \bar{E}} w_i - 1 - 3|\bar{E}| + |\bar{V}|$.

Hence, from any solution to MLM-CFD on $\sigma_{\bar{H}}$, we get a set packing for $\bar{H}$ having maximum weight. $\square$

We now show the $\text{F}\Delta_2^{\text{P}}$-completeness. To this end, consider the $\exists$-BETTER problem that, given as input a scenario, an n-dimensional vector v, and a number w, asks whether there is an allocation $\mathbf{B}$ such that $\xi(\mathbf{B}) \prec v$, or $\xi(\mathbf{B}) = v$ and $\text{SW}(\mathbf{B}) > w$. This problem is clearly in NP, and it can be used as an oracle for computing a maxileximin allocation.

Theorem 6. MLM-CFD *is* $\text{F}\Delta_2^{\text{P}}$*-complete.*

Proof Sketch. After Theorem 4 and Theorem 5, we have to focus on the membership in $\text{F}\Delta_2^{\text{P}}$ only. The problem can be solved with a polynomial number of calls to an NP oracle for $\exists$-BETTER as follows. Let M be the maximum sum of the values of the goods V over all agents, which is an upper bound for any agent's envy, and let $M_{sw} = n * M$, which is an upper bound for the social welfare. Use binary search over the n elements of the vector of envies to compute the leximin vector of envies, say ξ^*: call the $\exists$-BETTER oracle with parameters (σ, v, M_{sw}), where v is the vector of envies $\langle M/2, 0, \ldots, 0\rangle$; if the answer is "yes" then replace $M/2$ by $M/4$ and try again, otherwise try with $3/4M$. Once the value for the first position have been found, say e_1, repeat the procedure to compute the minimum second worst-envy by calling the oracle with an envy-vector parameter $v = \langle e_1, M/2, 0, \ldots, 0\rangle$. At the end of this phase, use again logarithmic search to maximize the social welfare, by fixing the envy-vector parameter to ξ^* and changing the threshold value for the social welfare at each call of $\exists$-BETTER (starting with the parameters $(\sigma, \xi^*, M_{sw}/2)$. Finally, by using a classical self-reduction technique, compute an allocation with the desired envy profile and social welfare value by performing further $n \times |V|$ calls to an NP oracle. $\square$

Smooth Utility Functions

A class $\mathcal{C}$ of allocation scenarios has *smooth* utility functions if their output values are polynomially bounded, that is, if their maximum value-size M_σ is $O(\log(|N|+|V|))$, for each $\sigma \in \mathcal{C}$. Note that one can equivalently consider $O(\log(|N| \cdot |V|))$, which is in $O(\log(|N|+|V|)^2) = O(\log(|N|+|V|))$.

It is immediate to check that proofs of Theorem 4 and Theorem 5 provide us with NP-hardness results even when such functions are considered. Indeed, the problems considered in the reductions are well-known to be NP-hard even if nodes/hyperedges have unitary weight. In fact, we can show that MLM-CFD remains complete for $\mathrm{F}\Delta_2^{\mathrm{P}}$ even on smooth functions. This time, we need to use a different reduction with more involved graph topologies.

Theorem 7. MLM-CFD *is* $\mathrm{F}\Delta_2^{\mathrm{P}}$*-hard even on smooth utility functions.*

Proof Sketch. Consider the $\mathrm{F}\Delta_2^{\mathrm{P}}$-complete LEX-SAT problem (Krentel 1988): given a (w.l.o.g.) satisfiable Boolean formula $\Phi = c_1 \wedge \cdots \wedge c_m$ in conjunctive normal form over the set $\{\alpha_1, ..., \alpha_n\}$ of variables, compute the *lexicographically maximum satisfying assignment*, with variables being ordered by their indices (that is, α_1 is the most significant variable). Based on Φ, we build (in polynomial time) an allocation scenario $\sigma_\Phi = (N, G, \{u_i\}_{i \in N})$ as follows. The graph $G = (V, E)$ is defined over the set $V = \{\alpha_p, \bar{\alpha}_p \mid p \in \{1, ..., n\}\} \cup \{c_1, ..., c_m\} \cup \{e, h\}$. Its edges are such that: for each $p \in \{1, ..., n\}$, E contains the edges $\{\alpha_p, \bar{\alpha}_p\}$, $\{\alpha_p, e\}$, and $\{\bar{\alpha}_p, e\}$; for each clause c_j and variable α_p occurring positively (resp., negatively), E contains $\{c_j, \alpha_j\}$ (resp., $\{c_j, \bar{\alpha}_j\}$); E contains the edge $\{e, h\}$, and no further edge is in E. As an example, Figure 4 reports the graph associated with the scenario σ_Φ, for the formula $\Phi = (\alpha_1 \vee \neg\alpha_2 \vee \alpha_3) \wedge (\alpha_4)$.

The set N of agents in σ_Φ is such that:

- N contains AG(h) such that $u_{\mathrm{AG}(h)}(h) = m(n+1)$ and $u_{\mathrm{AG}(h)}(v) = 0$, for each $v \in V \setminus \{h\}$;
- N contains AG(e) such that $u_{\mathrm{AG}(e)}(h) = m(n+1)$ and $u_{\mathrm{AG}(e)}(c_j) = (n+1)$, for each $j \in \{1, ..., m\}$, and $u_{\mathrm{AG}(h)}(v) = 0$, for each $v \in V \setminus \{c_1, ..., c_m\}$;
- N contains AG$(1), ...,$ AG(n) such that $u_{\mathrm{AG}(p)}(\bar{\alpha}_p) = n - p + 1$, for each $p \in \{1, ..., n\}$, and $u_{\mathrm{AG}(h)}(v) = 0$, for each $v \in V \setminus \{\bar{\alpha}_1, ..., \bar{\alpha}_p\}$;
- no further agent is in N.

We now claim that there is a one-to-one correspondence between leximin allocations for σ_Φ and lexicographically maximum satisfying assignments for Φ. The key ingredient to prove the claim is that any leximin allocation $\mathbf{B}$ is such that $\mathbf{B}_{\mathrm{AG}(h)} \supseteq \{h\}$ and $\mathbf{B}_{\mathrm{AG}(e)} \supseteq \{e, c_1, ..., c_m\}$. In particular, by the connectedness condition of $\mathbf{B}_{\mathrm{AG}(e)}$, it holds that the assignment $\tau_{\mathbf{B}}$ such that $\tau_{\mathbf{B}}(\alpha_p) = \texttt{true}$ iff $\mathbf{B}_{\mathrm{AG}(e)} \supseteq \{\alpha_p\}$ is satisfying—and the construction is possible since Φ is satisfiable, so that we can always build $\mathbf{B}$ in a way that AG(h) and AG(e) do not envy any other agents. Eventually, we have just to observe that the envy of agent AG(p) is $n - p + 1$ if $\bar{\alpha}_p \in \mathbf{B}_{\mathrm{AG}(e)}$ holds; otherwise, AG(p) does not envy any other agent. That is, the vector of the envies correspond to the variables that evaluate to true in the associated assignment.

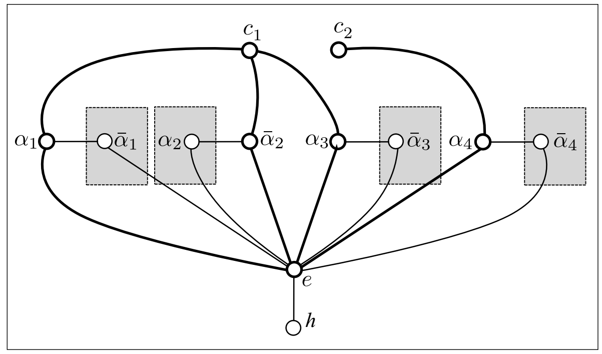


Figure 4: Construction in the proof of Theorem 7.

For the other way round, note that any truth assignment τ immediately identifies an allocation $\mathbf{B}$ with $\tau_{\mathbf{B}} = \tau$. $\square$

Bounded Number of Agents

We now focus on classes of allocation scenarios where the number of agents is bounded by some fixed constant. Observe that, if the underlying graphs are trees, then MLM-CFD can be solved in polynomial time by exhaustively enumerating all possible (polynomially many) allocations (cf., Bouveret et al. 2017). It is thus natural to consider classes of graphs that generalize acyclicity, as the quasi-acyclic graphs formalized by the notion of treewidth (Robertson and Seymour 1986).

Let $G = (V, E)$ be an undirected graph, with V and E being its (non-empty) sets of vertices and edges, respectively. A *tree decomposition* of G is a pair $\langle T, \chi \rangle$, where T is a tree, and χ is a labeling function assigning to each vertex p in T a set of nodes $\chi(p) \subseteq V$, such that the following conditions are satisfied: (1) for each node $x \in V$, there exists p in T such that $x \in \chi(p)$; (2) for each edge $\{x, y\} \in E$, there exists p in T such that $\{x, y\} \subseteq \chi(p)$; and, (3) for each node $x \in V$, the subgraph of T induced by all vertices p such that $x \in \chi(p)$ is connected.

The *width* of $\langle T, \chi \rangle$ is the number $\max_{p \in T}(|\chi(p)| - 1)$. The *treewidth* of G, denoted by $tw(G)$, is the minimum width over all its decompositions. Treewidth is a generalization of acyclicity: G is acyclic if, and only if, $tw(G) = 1$.

Arbitrary Utility Functions

We first point out that, looking for islands of tractability, it is crucial to consider restrictions of utility functions.

Indeed, MLM-CFD is NP-hard even for classes with only two agents having the same utility functions and goods graphs having bounded treewidth: we can show that even deciding whether there is an envy-free allocation is NP-hard, in these cases. The proof is routine and uses a reduction from the PARTITION problem (Garey and Johnson 1979).

Input: $\sigma, E, \langle T, \chi \rangle$
Task: decide whether there is an allocation $\mathbf{B}$ such that, $\forall i, j \in N$, $E[i][j] = u_i(\mathbf{B}_j)$
Method:
  let q be $root$ of T
  CHECKPROFILE$(q, \emptyset, E, \emptyset, N)$

Procedure CHECKPROFILE$(q, \lambda_p, \omega_p, CT_p, rest)$
Begin Procedure
  guess a (total) mapping $\lambda_q : \chi(q) \mapsto N$ assigning to agents the goods occurring at q
  let $new = AG_q \setminus AG_p$ (the new active agents introduced at q)
  `Stop and Fail if` $new \not\subseteq rest$ or $\lambda_q(g) \neq \lambda_p(g)$, for some good g occurring in both mappings
  let $terminal = AG_p \setminus AG_q$ (the agents that are active at p and disappear at q)
  for each agent $a \in terminal$, `Stop and Fail` **if**
    $\omega_p[i][a] \neq 0$, for some $i \in N$ (some value of the bundle assigned to a does not match the profile E), `or`
    $CT_p[a]$ contains more than one component (the bundle of goods assigned to a is not connected)
  for each agent $a \in AG_q$:
    $\forall i \in N$, update the profile value for i w.r.t. the bundle assigned to a: $\omega_q[i][a] = \omega_p[i][a] - \sum_{g \in (goods_q[a] \setminus goods_p[a])} u_i(g)$
    if $a \in new$ **then guess** the components-tree $CT_q[a]$ with the connected components of the induced subgraph $G[goods_q[a]]$
    else guess an update of the components-tree $CT_q[a]$ (to consider new goods and remove disappeared goods)
  if q is a leaf of T **then**
    if $rest \neq new$ **then** `Stop and Fail` (there is some agent without any bundle of goods)
    for each agent $a \in AG_q$ and **each** agent $i \in N$, `Stop and Fail` **if**
      $\omega_q[i][a] \neq 0$ (the profile value for i w.r.t. the bundle assigned to a does not match $E[i][a]$), or
      $CT_q[a]$ contains more than one component (the bundle of goods assigned to a is not connected)
  else (q is not a leaf of T)
    guess a partition $rest', rest''$ of the agents in $rest \setminus new$ to be dealt with in the two subtrees
    guess two matrices ω'_q and ω''_q such that, $\forall i \in N, \forall a \in AG_q, \omega'_q[i][a] + \omega''_q[i][a] = \omega_q[i][a]$, and
      $\forall i \in N, \forall j \in N \setminus AG_q, \omega'_q[i][j] = \omega''_q[i][j] = \omega_q[i][j]$
    guess a split $(CT'_q[a], CT''_q[a])$ of the components-tree of each $a \in AG_q$
    Let ℓ and r be the left child and the right child of q in T
    CHECKPROFILE$(\ell, \lambda_q, \omega'_q, CT'_q, rest')$
    CHECKPROFILE$(r, \lambda_q, \omega''_q, CT''_q, rest'')$
End Procedure

Figure 5: Deciding the existence of an allocation matching a given value profile

Smooth Utility Functions

Finally, we are able to provide our main tractability result, for all those instances where there is a bounded number of agents possibly competing for a large number of goods, whose connections have a small degree of cyclicity, and where utility functions only use small values (that is, values at most polynomially larger than the input size, so that they have a logarithmic encoding).

Theorem 8. *On classes of smooth allocation scenarios with a bounded number of agents and having bounded treewidth,* MLM-CFD *belongs to* $\mathrm{L}^{\mathrm{LogCFL}}$*, and thus it is a parallelizable and polynomial time problem.*

Proof Sketch. Let σ be a smooth allocation scenario over a graph $G = (V, E)$ with $|N| = k$ agents, and denote its size by $\|\sigma\|$. We show that the algorithm described in the proof of Theorem 6 can be implemented by a LOGSPACE machine with a LogCFL oracle. First observe that, because we have k agents and logspace utilities, the binary search procedure requires logspace only. It then suffices to show that $\exists$-BETTER is in LogCFL. To this end, for any allocation $\mathbf{B}$, consider the matrix E such that $E[i][j] = u_i(B_j)\ \forall i, j \in N$, which we call the value profile of $\mathbf{B}$. Note that it is immediate to compute, based on E, the envy vector for the agents, call it $\xi(E)$, as well as the social welfare $\sum_{i \in N} E[i][i]$. Then, for an instance (σ, v, s) for $\exists$-BETTER, where v is a vector of (non-increasing) envies and s is a threshold for the social welfare, we can enumerate in polynomial time all value profiles E such that $\xi(E) \prec v$, or $\xi(E) = v$ and $\sum_{i \in N} E[i][i] > s$. Indeed, even if there are exponentially many bundle allocations, there are only polynomially-many distinct value profiles, whose sizes are logarithmic—as we deal with k^2 small values. We thus get the tractability for $\exists$-BETTER by exhibiting the algorithm shown in Figure 5 that, given any value profile E, checks whether there actually exists an allocation whose value profile is equal to E. The algorithm is based on a non-deterministic Boolean function CHECKPROFILE that can be implemented on a *logspace Alternating Turing Machine (ATM)*, with polynomial accepting computation trees, which entails that the problem belongs to LogCFL. The ATM works top-down along a tree decomposition. In particular, because the treewidth is bounded by a fixed constant, we can compute beforehand a minimum-width tree decomposition of the graph G in linear time (Bodlaender 1993)—or we can compute it when needed, as the problem belongs to $\mathrm{L}^{\mathrm{LogCFL}}$, too. Moreover, we can transform this decomposition into a tree decomposition $\langle T, \chi \rangle$ of G having the same width and such that T is a full binary tree.

High-level description. We use "node" to refer to any node of the decomposition tree T, while "good" refers to any node of G. If $\pi : X \mapsto Y$ is a partial mapping from X to Y, then we write $x \in \pi$ to mean any element x in the active domain of π, that is, such that $x \in X$ is actually mapped to some $\pi(x) \in Y$. Moreover, $\emptyset$ denotes the empty mapping (having an empty active domain). We describe the ATM as a high-level recursive procedure performing non-deterministic *guess* operations. More precisely, when called with a node q as its parameter, the procedure CHECKPROFILE starts *guessing* an assignment of goods $\lambda_q : \chi(q) \mapsto N$. Denote by AG_q the so-called *active agents* at q, that is, the set of agents $\{a \in N \mid \lambda_q(g) = a, \text{ for some } g \in \chi(q)\}$ to which the goods occurring at q are assigned. Denote by $goods_q[a]$ the goods assigned to such an agent a at q, i.e., the set $\{g \in \chi(q) \mid \lambda_q(g) = a\}$. Let t be the treewidth of G, and note that there are at most $t + 1$ goods in $\chi(q)$, and thus at most $t+1$ active agents at q. These sets and the mapping can be stored with $O(\log \|\sigma\|)$ bits (goods, agents, and nodes are encoded with suitable pointers to the input tape, or indices identifying such elements). The machine also stores a matrix ω_q which holds, for each pair $i, j \in N$, the value $\omega_q[i][j]$ that is still missing in the evaluation of agent i of the (partial) bundles assigned to agent j, in order to match the desired entry $E[i][j]$ of the value-profile. At the first call of the function at the root of the decomposition tree, this matrix is set to E, and each entry, identified by a pair $i, j \in N$, is then reduced during the computation by considering the values according to u_i of the goods assigned to j along the decomposition tree. Before the recursive calls on the children of q in T, the ATM *guesses* two matrices ω'_q and ω''_q that encode the values that, for each entry, should be dealt with in the two subtrees rooted at the children of q. Moreover, it stores the set of agents $rest$ whose bundles have to be assigned in the subtree rooted at q; at the root, $rest$ is set to N. Before performing the recursive calls, the ATM *guesses* a partition $rest'$ and $rest''$ of $rest \setminus new$ holding the agents not yet considered (in the set of new active agents new) that should be dealt with in the subtrees rooted at the children of q. At leaves, all agents must be considered.

Connection constraint. The more involved issue is how the ATM can check that the bundle $\mathbf{B}_a$ of goods to be assigned to agent a is connected, and that no good is assigned to different agents during the algorithm, possibly at different non-adjacent nodes of the decomposition tree T. This is non-trivial, as we do not have enough memory to store bundles at vertices of T (recall that we can use just logarithmic space and so we can store a constant number of indices of goods, while each bundle may contain an arbitrary number of goods). Note, on the one hand, that the nodes of the decomposition tree T containing goods of $\mathbf{B}_a$ induce a connected subtree of T. But, on the other hand, those specific goods $goods_q[a]$ from $\mathbf{B}_a$ that occur at node q of this subtree can be goods that are not directly connected in the graph G. Because of the logarithmic-space constraint, we may have only a partial view of $\mathbf{B}_a$, and we cannot say anything about the connection property of $\mathbf{B}_a$.

The main ingredient here is to manage, at any node q of the decomposition-tree, a logarithmic-space partial representation of a spanning tree of the bundle assigned to each active agent a at q, denoted by $CT_q[a]$ and called *components-tree*. The vertices of $CT_q[a]$ encode a partition of the goods $goods_q[a]$ assigned to a at q, and are initially set to be the connected components of the subgraph $G[goods_q[a]]$ induced on G by these goods. Note that there are at most t such vertices, because the treewidth is t and at most $t + 1$ goods occur at q. The edges (at most t) of any components-tree are *guessed* non-deterministically by the ATM, and encode paths among the connected components. These paths involve goods, still unknown at this point of the algorithm, that will eventually be found while traversing the decomposition tree. Any edge between two components in $CT_q[a]$ encodes a placeholder that we must find somewhere down in T an actual connection between some pair of goods occurring in these components. Let ℓ and r be the children of q, we require that the actual presence of such a connection is checked either in the subtree rooted at ℓ or in the subtree rooted at r. If two components are not to be checked in a subtree, they are combined in one component (the other subtree must eventually deal with them, by checking the existence of the required path). To this end, a suitable *split operation* produces a pair of components-tree $CT'_q[a]$ and $CT''_q[a]$ to be checked recursively.

Recursive calls. Eventually, CHECKPROFILE executes a recursive call for each child of q in order to check that the non-deterministic choices performed at q are actually correct. To conclude, just observe that all the information needed at each call of CHECKPROFILE can be encoded in logarithmic space, as required. □

Conclusion

We have proposed maxileximin allocations as a fair method for allocating indivisible goods and have examined their computational properties, by identifying both hard and easy instances. Our approach guarantees the existence of a solution, which is a critical feature in any application. Moreover, in our notion, efficiency is intertwined with fairness. Indeed, it is easy to show examples of envy-free solutions in which almost all agents are dissatisfied, while other envy-free solutions offer substantially improved outcomes for all agents. We firmly believe that, in such situations, choosing the envy-free solution that maximizes agents' utilities is a matter of fairness, not just of system efficiency.

The techniques described in the paper can be used to demonstrate the tractability of different measures of dissatisfaction, such as minimizing the Ordered Weighted Average (OWA) of the envy vector (Shams et al. 2021). Exploring islands of tractability for scenarios with an arbitrary number of agents is a potential area for future research. Additionally, it is worth noting that our approach can be easily extended to incorporate further refinements of leximin allocations. This could involve considering additional measures to optimize in agents' dissatisfaction profiles (currently focused on envies alone), and potentially incorporating other efficiency notions (we have examined social welfare). Specifically, it would be interesting to study the problem of computing MLM allocations where we additionally consider metrics related to the notion of *maximin share* (Budish 2011).

Acknowledgements

This work was funded by the following two projects under the NextGenerationEU NRRP MUR program: FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, and Tech4You - Technologies for climate change adaptation and quality of life improvement (ECS0000009), call for the creation and strengthening of 'Innovation Ecosystems', building 'Territorial R&D Leaders' (Directorial Decree n. 2021/3277).

References

Amanatidis, G.; Aziz, H.; Birmpas, G.; Filos-Ratsikas, A.; Li, B.; Moulin, H.; Voudouris, A. A.; and Wu, X. 2023. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.*, 322: 103965.

Amanatidis, G.; Birmpas, G.; and Markakis, V. 2018. Comparing Approximate Relaxations of Envy-Freeness. In *Proc. of IJCAI'18*, 42–48.

Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018. Finding Fair and Efficient Allocations. In *Proc. of EC'18*, 557––574.

Bei, X.; Igarashi, A.; Lu, X.; and Suksompong, W. 2022. The Price of Connectivity in Fair Division. *SIAM J. Discret. Math.*, 36(2): 1156–1186.

Bei, X.; Lu, X.; Manurangsi, P.; and Suksompong, W. 2021. The Price of Fairness for Indivisible Goods. *Theory Comput. Syst.*, 65(7): 1069–1093.

Bilò, V.; Caragiannis, I.; Flammini, M.; Igarashi, A.; Monaco, G.; Peters, D.; Vinci, C.; and Zwicker, W. S. 2022. Almost envy-free allocations with connected bundles. *Games Econ. Behav.*, 131: 197–221.

Bodlaender, H. L. 1993. A Linear Time Algorithm for Finding Tree-decompositions of Small Treewidth. In *Proc. of STOC'93*, 226–234.

Bouveret, S.; Cechlárová, K.; Elkind, E.; Igarashi, A.; and Peters, D. 2017. Fair Division of a Graph. In *Proc. of IJCAI'17*, 135–141.

Bouveret, S.; Cechlárová, K.; and Lesca, J. 2018. Chore division on a graph. *Autonomous Agents and Multi-Agent Systems*, 1–24.

Bouveret, S.; Chevaleyre, Y.; Maudet, N.; and Moulin, H. 2016. *Fair Allocation of Indivisible Goods*, 284–310. Cambridge University Press.

Brams, S. J.; and Taylor, A. D. 1996. *Fair division: From cake-cutting to dispute resolution*. Cambridge University Press.

Budish, E. 2011. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, 119(6): 1061–1103.

Cai, Q.; Filos-Ratsikas, A.; and Tang, P. 2016. Facility Location with Minimax Envy. In *Proc. of IJCAI'16*, 137—-143.

Caragiannis, I.; Kaklamanis, C.; Kanellopoulos, P.; and Kyropoulou, M. 2009. On Low-Envy Truthful Allocations. In Rossi, F.; and Tsoukias, A., eds., *Proc. of ADT'09*, 111–119.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2016. The Unreasonable Fairness of Maximum Nash Welfare. In *Proc. of EC'16*, 305–322.

Chandra, A. K.; Kozen, D. C.; and Stockmeyer, L. J. 1981. Alternation. *J. ACM*, 28(1): 114–133.

de Keijzer, B.; Bouveret, S.; Klos, T.; and Zhang, Y. 2009. On the Complexity of Efficiency and Envy-Freeness in Fair Division of Indivisible Goods with Additive Preferences. In Rossi, F.; and Tsoukias, A., eds., *Proc. of ADT'09*, 98–110.

Deligkas, A.; Eiben, E.; Ganian, R.; Hamm, T.; and Ordyniak, S. 2021. The Parameterized Complexity of Connected Fair Division. In *Proc. of IJCAI'21, Virtual Event / Montreal, Canada, 19-27 August 2021*, 139–145. ijcai.org.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co. ISBN 0716710447.

Goldberg, P.; Hollender, A.; and Suksompong, W. 2020. Contiguous Cake Cutting: Hardness Results and Approximation Algorithms. *J. Artif. Intell. Res.*, 69: 109–141.

Gottlob, G.; Leone, N.; and Scarcello, F. 2002. Computing LOGCFL certificates. *Theor. Comput. Sci.*, 270(1-2): 761–777.

Igarashi, A.; and Peters, D. 2019. Pareto-Optimal Allocation of Indivisible Goods with Connectivity Constraints. *Proceedings of AAAI'19*, 33(01): 2045–2052.

Krentel, M. W. 1988. The complexity of optimization problems. *JCSS*, 36(3): 490–509.

Lang, J.; and Rothe, J. 2016. Fair Division of Indivisible Goods. In *Economics and Computation*, 493–550. Springer.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On Approximately Fair Allocations of Indivisible Goods. In *Proc. of EC'04*, 125–131.

Lonc, Z.; and Truszczynski, M. 2018. Maximin Share Allocations on Cycles. In *Proc. of IJCAI'18*, 410–416.

Netzer, A.; Meisels, A.; and Zivan, R. 2016. Distributed Envy Minimization for Resource Allocation. *Autonomous Agents and Multi-Agent Systems*, 30(2): 364—-402.

Nguyen, T. T.; and Rothe, J. 2014. Minimizing envy and maximizing average Nash social welfare in the allocation of indivisible goods. *Disc. Appl. Mathematics*, 179: 54–68.

Robertson, N.; and Seymour, P. 1986. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3): 309–322.

Schmeidler, D. 1969. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17: 1163–1170.

Shams, P.; Beynier, A.; Bouveret, S.; and Maudet, N. 2021. Minimizing and Balancing Envy Among Agents Using Ordered Weighted Average. In *Proc. of ADT'21, Toulouse, France, November 3-5, 2021*, volume 13023 of *Lecture Notes in Computer Science*, 289–303. Springer.

Shams, P.; Beynier, A.; Bouveret, S.; and Maudet, N. 2022. Fair in the Eyes of Others. *J. Artif. Intell. Res.*, 75: 913–951.

Steinhaus, H. 1948. The problem of fair division. *Econometrica*, 16: 101–104.

Suksompong, W. 2017. Fairly Allocating Contiguous Blocks of Indivisible Items. In *Algorithmic Game Theory*, 333–344.

Suksompong, W. 2021. Constraints in fair division. *SIGecom Exch.*, 19(2): 46–61.

Walsh, T. 2020. Fair Division: The Computer Scientist's Perspective. In *Proc. of IJCAI'20*, 4966–4972. ijcai.org.

Weymark, J. A. 1981. Generalized Gini inequality indices. *Mathematical Social Sciences*, 1(4): 409–430.