ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230284

High-Multiplicity Fair Allocation Using Parametric Integer Linear Programming

Robert Bredereck[a,d], Andrzej Kaczmarczyk[b,d;*], Dušan Knop[c,d] and Rolf Niedermeier[d]

[a]Institut für Informatik, TU Clausthal, Clausthal-Zellerfeld, Germany
[b]AGH University, Kraków, Poland
[c]Czech Technical University in Prague, Prague, Czech Republic
[d]Technische Universität Berlin, Berlin, Germany
ORCiD ID: Robert Bredereck https://orcid.org/0000-0002-6303-6276,
Andrzej Kaczmarczyk https://orcid.org/0000-0003-1401-0157, Dušan Knop https://orcid.org/0000-0003-2588-5709,
Rolf Niedermeier https://orcid.org/0000-0003-1703-1236

Abstract. Using insights from parametric integer linear programming, we improve the work of Bredereck et al. [Proc. ACM EC 2019] on high-multiplicity fair allocation. Answering an open question from their work, we proved that the problem of finding envy-free Pareto-efficient allocations of indivisible items is fixed-parameter tractable with respect to the combined parameter "number of agents" plus "number of item types." Our central improvement, compared to their result, is to break the condition that the corresponding utility and multiplicity values have to be encoded in unary, which is required there. Concretely, we show that, while preserving fixed-parameter tractability, these values can be encoded in binary. Thus, we substantially expand the range of feasible utility and multiplicity values.

1 Introduction

Fairly allocating (indivisible) items [11] is a key issue in a world of limited resources, which is, for instance, reflected by multiple application contexts such as distributing food by food banks [41], university course assignment problems [18], or sharing computing resources [26]. In recent decades, studying fair allocation issues through the computational lens or, more generally, applying computer science toolbox [42] proved useful in advancing our knowledge of how to deal with finding desirable allocations. Examples include popular tools such as the Adjusted Winner Procedure [14] or the web platform spliddit.org [27] to name a few.

In this work, we focus on the so-called "high-multiplicity fair allocation" scenario in which various item types come in multiple copies. To understand important facets of our research contribution, let us, however, become more precise on the studied problem and the most relevant existing results.

We consider a set of item types, each coming with the number of actual items of this type, and a set of agents who report their non-negative utilities over each item type. An allocation of items is an assignment of disjoint sets of the items, called bundles, to the agents. In our work we first focus on one of the most prominent fairness concepts which is *envy-freeness*. It considers an allocation as *fair* if there is no agent that would prefer a bundle of any other agent over her own one. However, it is trivial to achieve envy-freeness by giving every agent an empty bundle. To circumvent this issue, several "efficiency" measures of allocations have been proposed. A very important one, *Pareto-efficiency*, requires that for an efficient allocation there exists no other allocation that is preferred by at least one agent and, at the same time, does not make any agent worse off. Combining the aforementioned concepts together, we end up with so-called *envy-free Pareto-efficient allocations* on which we mostly focus in this paper.

Finding envy-free Pareto-efficient allocations is a computationally very hard problem. For instance, the corresponding decision problem is Σ_2^P-complete for general utilities [13]. The hardness holds even for (positive) additive utilities [32]—here, the utility that an agent gets from a bundle is a sum of utilities that this agent reports for every item in the bundle. This model, due to its simplicity is frequently assumed in the scientific social choice literature [12, 14, 39] and also forms an important part of experimental studies [15, 21]. Notably, practically relevant tools (like the Adjusted Winner Procedure and the web platform[1] spliddit.org [27]) make use of additive utilities too.

Motivated by a high practical relevance of the problem of finding envy-free Pareto-efficient allocations assuming additive utilities, Bliem et al. [10] studied its fine-grained computational complexity providing several parameterized-tractability results. However, they left open a question whether the subject problem is fixed-parameter tractable with respect to the (combined) parameter "number of agents plus number of item types."[2] The question was then answered partially positively (with the restriction of unary encoded item multiplicities and utilities) in the work of Bredereck et al. [16].

Our Contribution Our main contribution is to strengthen the previous result of Bredereck et al. [16] by providing an algorithm offering

* Corresponding Author. Email: andrzej.kaczmarczyk@agh.edu.pl

[1] The spliddit.org webpage is currently (April 2023) unavailable. However, a github repository with the software is available at https://github.com/jogo279/spliddit.

[2] Technically, the open question was formulated for the parameter $n + u_{\text{diff}}$, where u_{diff} denotes the number of different values in the utility functions. This parameter can easily be seen to be equivalent to our parameter $n + m$ in terms of fixed-parameter tractability. Note that Bliem et al. [10] used the variable name m for the number of items and showed fixed-parameter tractability for this parameter.

better computational complexity lower-bound guarantees for finding envy-free Pareto-efficient allocations. To this end, applying techniques from parametric integer linear programming, we generalize their fixed-parameter tractability result regarding the parameterization by the number of agents and the number of item types. Specifically, we relax the requirement of unary encoded item multiplicities and utilities thereby allowing binary encodings.

Our result expands the range of values that we can deal with efficiently in the case of small numbers of agents and item types. Arguably, the case is quite relevant in practice, as all scenarios in the experimentally studied [15] data from spliddit.com [27] mostly considered at most 8 agents and 10 item types (with very few instances having at most 15 agents and 30 item types). Additional examples could include stock inheritance. Here, a portfolio consisting of around 30 companies (item types) is commonly advised by the experts. As the portfolio value grows, the number of share units (item multiplicities) of each company to share in the inheritance process can easily reach thousands. For such scenarios, algorithms guaranteeing fixed-parameter tractability for binary encoding of item multiplicities are a better bet for obtaining practically relevant running times than algorithms assuming unary encoding.

Furthermore, similarly to their result, our technique is applicable to a broad family of allocation problems emerging from different desiderata chosen to represent fairness (e.g., (group) envy-freeness, (group) envy-freeness up to one good, (group) envy-freeness up to any good, maximin share) and efficiency (e.g., completeness, welfare maximization, group Pareto-efficiency).

Overall, providing our result, we mainly contribute to the improvement of algorithmic tools allowing for searching *provably* fair and efficient allocations of indivisible items. Notably, our technique does not only allow for answering the question of the existence of fair and efficient allocations but it outputs such an allocation if it exists.

1.1 Related Work

Our work brings together the two worlds of fair allocations and parameteric Integer Linear Programs. Hence, we split the discussion of the related work into two parts organized thematically. We note that due to a flurry of literature dealing with fair allocations, we only focus on the works most relevant to ours.

Efficient and Envy-free Allocations of Indivisible Resources. Bouveret and Lang [13] were the first to study the computational complexity of computing Pareto-efficient and envy-free allocations of indivisible items in a systematic way. Their findings include Σ_2^P-completeness for the so-called monotonic dichotomous preferences as well as NP-hardness and polynomial-time solvability for several special cases. Most relevant to our setting with additive utility-based preferences, they showed that even if there are just two agents or if every agent assigns either utility value 0 or 1 to each item, the problem of finding a Pareto-efficient and envy-free allocation remains NP-hard. Moreover, de Keijzer et al. [32] showed that Σ_2^P-completeness even holds for positive additive preferences. Bliem et al. [10] analyzed the parameterized complexity, showing that the problem becomes tractable for the parameter "number of items" and various special settings but remains intractable for the parameter "number of agents."

Multiple approaches have been developed to relax fairness concepts in order to circumvent computational intractability as well as possible non-existence of Pareto-efficient and envy-free allocations. For instance, Lipton et al. [36] considered the concept of envy-freeness up to one good (EF1). Herein, every agent compares its bundle with the bundles of all other agents and she is envious if any other bundle minus the most valuable item in there is better than her own bundle. Further studied concepts include envy-freeness up to any good (EFX) [19, 37], minimum envy [36], group envy-freeness, group Pareto-efficiency [2], or graph envy-freeness [1, 9, 17, 4]. Amanatidis et al. [3] provide a comparison of approximate or relaxed fairness notions.

Caragiannis et al. [19] showed how to compute an allocation that maximizes Nash welfare and thus yields Pareto-efficiency and EF1. Barman et al. [8] improved this result and developed an algorithm that computes an allocation that is Pareto-efficient and EF1 with pseudo-polynomial running time (being polynomial in the number of agents, the number of items, and the maximum utility). While a round-robin allocation of items can be used to obtain a complete EF1 allocation in polynomial time when all items have positive utilities, Aziz et al. [5, 6] have argued that this procedure fails when items may have negative utilities. Leaving the complexity of computing Pareto-efficient and EF1 allocation (when negative utilities are allowed) open, they showed that a complete EF1 allocation can be found in polynomial time even when items with negative utilities are present.

The setting of high-multiplicity items (where items come in multiple copies) deserves a separate treatment. Copies of items played an important role in the seminal work of Budish [18]. However, there each agent's bundle was assumed to have to at most a single copy of a given resource (this follows from the fact that the author was focusing on an assignment problem, like assigning students to courses). Later, Gafni et al. [24] proposed a framework for studying the existence of EFX allocations in this model. The setting where an agent can obtain more resources of the same type was, to the best of our knowledge, first considered by Bredereck et al. [16] (on whose work we improve on). They establish a theoretical ILP-based framework for computing various types of efficient and fair allocations. The framework was later implemented and tested on real-data by Bredereck et al. [15]. Implicitly, the high-multiplicity setting is also present in the work of Eiben et al. [22]. They study parameterized complexity of finding graph envy-free allocations considering a parameterization (among others) by the number of item-types. The high-multiplicity regime has also been reinvented by Gorantla et al. [28] in the context of studying the conditions under which EF1 allocations exist.

Parametric ILP Aplications. Eisenbrand and Shmonin [23, Theorem 4.2] gave an algorithm that, if the number of variables is fixed, solves the given instance of Parametric ILP (PILP) in polynomial time (we formally define PILP in the Preliminaries). Köppe et al. [34] showed that one can express the negation of bilevel integer programs (a family of certain linear programs) as PILP and used the result of Eisenbrand and Shmonin to obtain polynomial-time solvability of bilevel integer programs in some restricted cases.

To the best of our knowledge, Crampton et al. [20, Corollary 2.2] were the first to give an "interpretation" of the result of Eisenbrand and Shmonin [23] in terms of parameterized complexity analysis. More specifically, they showed membership in the complexity class FPT, that is, they showed a running time $f(p, n) \cdot |I|$ for an instance I of PILP provided that the coefficients of the matrix A are encoded in unary. Using this result Crampton et al. [20] initiated the parameterized study of the so-called *resiliency* problems (such as the RESILIENCY CLOSEST STRING problem).

Knop et al. [33] used the interpretation of Crampton et al. [20] to solve a decade-long-standing open question of FPT-membership of a variant of the BRIBERY problem in the field of elections manipulation. Recently, Bredereck et al. [16] also used the interpretation of Crampton et al. [20] in the context of fair allocation. More specifically, they showed [16, Corollary 5] that finding a fair and efficient allocation is fixed-parameter tractable for few agents and few item types. The result holds for numerous different concepts of fairness and efficiency. Yet, their result holds only when the maximum utility value an agent assigns to an item type and item multiplicities are encoded in unary. As we shall shortly see, we are improving upon this result by allowing item multiplicities to be encoded in *binary*.

1.2 Organization

In the following Section 2, we first give necessary notation and formal preliminaries regarding allocations, parameterized complexity, and parameterized integer linear programs. Then, in Section 3, we lay foundations for proving our main result by presenting a convenient interpretation of Theorem 4.2 from the work of Eisenbrand and Shmonin [23] (our interpretation is more detailed than the one provided by Crampton et al. [20]). We proceed with formally stating our result and proving it in Section 4. Later, in Section 5 we discuss how to extend our main result to cover multiple further prominent fairness and efficiency concepts. In the last section (Section 6) we give conclusions.

2 Preliminaries

For a positive integer n, by $[n]$ we denote the set $\{1, 2, \ldots, n\}$. We use boldface letters, like $\mathbf{x}, \mathbf{y}$, to represent vectors. A vector $\mathbf{x}$ consisting of n coordinates is said to be in n dimensions or n-dimensional and we denote its i-th coordinate, $i \in [n]$, by x_i. For two vectors $\mathbf{x}$ and $\mathbf{y}$ in dimensions $n_{\mathbf{x}}$ and $n_{\mathbf{y}}$ respectively, vector $(\mathbf{x}, \mathbf{y})$ is a $(n_{\mathbf{x}} + n_{\mathbf{y}})$-dimensional vector $(x_1, \ldots, x_{n_{\mathbf{x}}}, y_1, \ldots, y_{n_{\mathbf{y}}})$. We symbolically denote some real matrix A with n rows and m columns by $A \in \mathbb{R}^{n \times m}$. We treat n-dimensional vectors as matrices with n rows and 1 column. A *polyhedron* is an intersection of half-spaces, that is, for some dimensions m and n, a polyhedron is a set $\{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ of vectors, for some $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Similarly, assuming the same notation and defining $\bar{A}$ and $\bar{\mathbf{b}}$ analogously, a *partially open polyhedron* is an intersection of half-spaces and open half-spaces, that is, a set $\{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b},\ \bar{A}\mathbf{x} < \bar{\mathbf{b}}\}$ of vectors.

2.1 Allocations, Envy-Freeness, and Pareto-Efficiency

Consider a set $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ of n agents, a set $\mathcal{I} = \{1, 2, \ldots, m\}$ of m item types with multiplicities m_i for each item $i \in \mathcal{I}$. An allocation $\boldsymbol{\pi}$ is an integral $(n \cdot m)$-dimensional vector $\boldsymbol{\pi} = \left(\pi^1_{a_1}, \ldots, \pi^1_{a_n}, \pi^2_{a_1}, \ldots, \pi^m_{a_n}\right)$, whose entries describe for each agent how many items of each item type are allocated to the agent. For each agent $a \in \mathcal{A}$, let $u_a \colon \mathcal{I} \to \mathbb{Z}$ be the agent's utility function (in fact, utility values may be rational numbers, in which case an equivalent problem instance with integral values can be obtained without loss of generality by multiplying each values by the least common multiplier of the denominators). We assume the preferences of the agents to be additive, which means that the utility value for a set of items is the sum of the items utility values. Thus, we define the satisfaction of agent $a \in \mathcal{A}$ from allocation $\boldsymbol{\pi}$ as $\sum_{i \in \mathcal{I}} u_a(i) \cdot \pi^i_a$; for brevity, we slightly abuse the notation and denote it by $u_a(\boldsymbol{\pi})$.

Before we proceed, let us fix a set $\mathcal{A}$ of n agents and a set $\mathcal{I}$ of m item types with multiplicities m_i for each item type $i \in \mathcal{I}$. Let $\boldsymbol{\pi}$ be an allocation of the items to the agents in $\mathcal{A}$. In the following two definitions we provide formal phrasings of envy-freeness and Pareto-efficiency, which play a central role in our study.

Definition 1. An allocation $\boldsymbol{\pi}$ of the items $\mathcal{I}$ with multiplicities m_i, $i \in \mathcal{I}$, to agents $\mathcal{A}$ is *envy-free* if there is no two agents $a \in \mathcal{A}$ and $a' \in \mathcal{A}$ such that $u_a(\boldsymbol{\pi}) < \sum_{i \in \mathcal{I}} u_a(i) \cdot \pi^i_{a'}$.

Definition 2. An allocation $\boldsymbol{\pi}$ of the items $\mathcal{I}$ with multiplicities m_i, $i \in \mathcal{I}$, to agents $\mathcal{A}$ is *Pareto-dominated* if there exists another allocation $\boldsymbol{\pi}'$ (over the same sets of agents and items together with their multiplicities) such that for every agent $a \in \mathcal{A}$ it holds that $u_a(\boldsymbol{\pi}') \geq u_a(\boldsymbol{\pi})$ and for at least one agent the inequality is strict. An allocation is *Pareto-efficient* if it is not Pareto-dominated.

In our work, we focus on a decision problem in which we ask whether for given sets of agents and resources, an allocation that is simultaneously envy-free and Pareto-efficient exists.

EEF–ALLOCATION

Input:	A set $\mathcal{A}$ of n agents, a set $\mathcal{I}$ of m item types, agent utilities $u_a \colon \mathcal{I} \to \mathbb{Z}$ for every $a \in \mathcal{A}$, and item multiplicities $m_i \in \mathbb{N}$ for each $i \in \mathcal{I}$.
Question:	Is there an envy-free Pareto-efficient allocation?

The name of the problem, standing for "efficient envy-free" allocation might be misleading in the light of the fact that in the literature "efficiency" has multiple embodiments (besides Pareto-efficiency, perhaps the most frequent ones are completeness or social welfare maximization). However, for clarity, we decided to keep the name as defined by Bouveret and Lang [13] and then consequently used by the follow-up works [10, 16].

2.2 Parameterized Complexity

A parameterized (decision) problem's input consists of a decision problem instance I and a parameter value k; the task is then to decide whether (I, k) is a "yes"-instance. We say that a parameterized problem is *fixed-parameter tractable with respect to* k (belongs to the class FPT with respect to k) if there is an algorithm deciding (I, k) in $f(k) \cdot \mathrm{poly}(|I|)$ time, where $|I|$ is the size of the input and $f(k)$ is an arbitrary computable function of parameter k. Intuitively, the exponential blow-up is then related only to the value of parameter k, which allows for efficient computation of the problem if k is small. The following proposition describing a relation between various functions values will come handy later.

Proposition 1 ([29, Lemma 3.10])**.** *For every two computable functions $g \colon \mathbb{N} \to \mathbb{N}$ and $h \colon \mathbb{N} \to \mathbb{N}$ with $g(n) = o(\log(n))$, there exists a computable function $f \colon \mathbb{N} \to \mathbb{N}$ such that for every k and n we have $2^{g(n)h(k)} \leq f(k) \cdot n$.*

2.3 Parametric Integer Programming

For a rational polyhedron $Q \subseteq \mathbb{Q}^{m+p}$, the *integer projection* of Q, denoted by $Q/\mathbb{Z}^p$, is a collection of all vectors $\mathbf{b} \in \mathbb{R}^m$ for which there exists an integral vector $\mathbf{z} \in \mathbb{Z}^p$ such that $(\mathbf{b}, \mathbf{z}) \in Q$. Thus, formally

$$Q/\mathbb{Z}^p := \{\mathbf{b} \in \mathbb{Q}^m \colon (\mathbf{b}, \mathbf{z}) \in Q \text{ for some } \mathbf{z} \in \mathbb{Z}^p\}\,.$$

Parametric Integer Programming (PILP) is the following problem. Given a matrix $A \in \mathbb{Q}^{m \times n}$ and a rational polyhedron $Q \subseteq \mathbb{Q}^{m+p}$, decide if for all vectors $\mathbf{b} \in \mathbb{Q}^m$ in the integer projection of Q, the system of inequalities $A\mathbf{x} \leq \mathbf{b}$ has an integral solution. In other words, one has to decide the validity of the sentence

$$\forall \mathbf{b} \in Q/\mathbb{Z}^p \; \exists \mathbf{x} \in \mathbb{Z}^n \colon \quad A\mathbf{x} \leq \mathbf{b}\,. \qquad \text{(PILP)}$$

Intuitively, PILP consists of a collection of integer linear programs defined by A and right-hand side vectors $\mathbf{b}$, where the latter ones come from the integer projection $Q/\mathbb{Z}^p$. The question then is whether each of these integer linear programs has some feasible solution. The PILP problem is complete for the class Π_2^p [40, 43].

3 Preparation for Main Result

We devote this section to describe important consequences resulting from the work of Eisenbrand and Shmonin [23, Theorem 4.1 and Theorem 4.2]. Most importantly, their results allow for efficiently solving PILP subject to additional constraints. As it will turn out, we are able to formulate EEF–ALLOCATION in a way that respects these constraints. Yet, before we show the formulation in Section 4, we discuss the aforementioned consequences in detail and present them formally in Proposition 2.

Despite the Π_2^p-completeness of the PILP problem, Eisenbrand and Shmonin [23, Theorem 4.1 and Theorem 4.2] gave a polynomial-time algorithm for PILP for the fixed number of variables and dimension n (their work extended the pioneering—to the best of our knowledge—works of Kannan [30, 31] on efficient algorithms for PILP). An analysis of their algorithm leads to the following Proposition 2; we discuss its details afterwards.

Proposition 2. *There is an algorithm deciding the sentence* (PILP) *in*

$$f(m, n, p) \cdot \phi^{h(n)} \cdot \mathrm{poly}(L)$$

time, where ϕ is the size (encoding length) of any column in A, L is the encoding length of the sentence and (the description of) the polyhedron Q, and f and h are computable functions. Moreover, if the sentence (PILP) *is not valid, then a certificate $\mathbf{b} \in Q$ is provided (i.e., $A\mathbf{x} \leq \mathbf{b}$ has no integral solution with such a $\mathbf{b}$).*

Proposition 2 essentially follows from an in-depth analysis of a known result [23, Theorem 4.2]. A similar investigation has also been provided by Crampton et al. [20]. However, we decided to slightly adjust it to our needs and hence we present it in more detail. Since Proposition 2 plays an important role in our result, we believe that discussing its argument explicitly is important for the completeness of our paper.

In the algorithm backing Proposition 2, we first utilize the Fourier–Motzkin elimination procedure to make sure that for all $\mathbf{b} \in Q$ the system $A\mathbf{x} \leq \mathbf{b}$ has a fractional solution. If this is not the case, then a corresponding vector $\mathbf{b}$ is reported which certifies the right-hand side vector for which the PILP sentence has no solution. Running this procedure for all $\mathbf{b} \in Q$ yielding the corresponding integer linear programs $A\mathbf{x} \leq \mathbf{b}$, requires solving $f'(m, n)$ many mixed integer linear programs in dimension p. This can be done in $p^{O(p)}\,\mathrm{poly}(L)$ time using Lenstra's celebrated result [35] about solving integer linear programs in bounded dimensions.

Second, we partition the polyhedron Q into t partially open polyhedrons S_i, $i \in [t]$. Due to a result by Eisenbrand and Shmonin [23, Theorem 4.1], the number t of partially open polyhedra S_i, $i \in [t]$, is expressed (using helper constants $\bar{\omega}(n)$ and $h(n)$, which we describe below) as

$$t = O\left(\left(m^{2n}\phi^{n-1}\right)^{n\bar{\omega}(n)}\right) = f''(m, n) \cdot \phi^{h(n)}\,.$$

Here, $\bar{\omega}(n) = \Pi_{i=1}^{n}\omega(n)$, where $\omega(n)$ is the constant from the flatness theorem (the current best value is $\omega(n) = O(n^{3/2})$ [7]), and $h(n) = n(n-1) \cdot \bar{\omega}(n)$. Importantly, Eisenbrand and Shmonin [23, Theorem 4.1] show that each S_i, $i \in [t]$, is an integer projection of some partially open polyhedron S_i', that is $S_i = S_i'/\mathbb{Z}^{\ell_i}$; additionally they show that $\ell_i = O(\bar{\omega}(n))$, $i \in [t]$. Lastly, the result of Eisenbrand and Shmonin [23, Theorem 4.1] gives, for each $i \in [t]$, a collection of $k_i = f'''(n)$ specific transformations T_{ij}, for $j \in [k_i]$. The transformations are very specific in the sense that for each $\mathbf{b} \in S_i$ there is an integral point in the polyhedron $P_{\mathbf{b}} := \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\}$ if and only if $T_{ij}(\mathbf{b}) \in P_{\mathbf{b}}$ for some $j \in [k_i]$. The negation of this condition can be verified using a mixed integer linear program for each $i \in [t]$; such an ILP has $(k_i + 1)n + \ell_i + p$ integral variables. It holds that if the input sentence (PILP) is not valid, then one of the above mixed ILPs is feasible; thus, again, providing the claimed certificate $\mathbf{b}$. Carefully inspecting the two parts of the above-sketched algorithm reveals that it runs in the requested $f(m, n, p) \cdot \phi^{h(n)} \cdot \mathrm{poly}(L)$ time.

4 Finding EEF–Allocations via PILP

The interpretation of Theorem 4.2 of Eisenbrand and Shmonin [23] presented in Section 2 contains an important bit. Specifically, we observed that it is possible to derive a certificate of infeasibility of a given PILP sentence. This inspired us to consider the following reasoning, which we employ to derive our result about finding envy-free and Pareto-efficient allocations. Instead of focusing directly on EEF–ALLOCATION, we decided to work with the complementary problem. This way, by obtaining the certificate of infeasibility for the complementary problem, we in fact get a (membership) certificate for the original problem. In more details, we think of a problem of deciding whether "every envy-free allocation is Pareto-dominated." If such a sentence is invalid, then a certificate proving it is an envy-free allocation that cannot be Pareto-dominated. It is worth pointing out that due to the certificate, we do not only answer the question posed by EEF–ALLOCATION but we also find an envy-free and Pareto-efficient allocation, which makes our approach constructive.

The method described above leads us to the main contribution of our work, which strengthens Corollary 5 of Bredereck et al. [16] about fixed-parameter tractability of EEF–ALLOCATION with respect to the combined parameter "number of agents plus number of items." Therein, the authors devise the negation of EEF–ALLOCATION in a similar spirit to ours (however, their approach is fundamentally different as it is based on analyzing a collection of improving steps among which none can be added to improve a given allocation) employing the big-M method to do so. We avoid this method, which (as used in the mentioned paper) forces a unary encoding of the input item multiplicities and utility values, arriving at our Theorem 1, which offers the same computational complexity guarantees but *does not* require the unary encoding of the discussed input elements.

Theorem 1. *Let I be an instance of the* EEF–ALLOCATION *problem with the maximum input utility value $u_{\max} = 2^{o(\log|I|)}$. Then, there is an algorithm that decides I in $f(m+n) \cdot \mathrm{poly}(|I|)$ time, for some computable function $f \colon \mathbb{N} \to \mathbb{N}$ and $|I|$ being the size of I.*

Before we proceed with proving Theorem 1 in the following Section 4.1, we remark that our technique also applies to other variants of EEF–ALLOCATION where we replace envy-freeness or Pareto-efficiency with related concepts. We devote a separate section (Section 5) to a detailed discussion about these additional applications.

4.1 Proving the result

Employing Proposition 2, we now show how to efficiently solve the EEF–ALLOCATION problem for the (combined) parameter "number of agents plus number item types," obtaining a proof of Theorem 1. From now on, we fix a set $\mathcal{A}$ of n agents and a set $\mathcal{I} = \{1, 2, \dots, m\}$ of m item types with multiplicities m_i, $i \in \mathcal{I}$.

As already discussed, we show the FPT-membership of EEF–ALLOCATION for the parameter $n + m$ by constructing a PILP sentence deciding whether every envy-free allocation of a given collection of items is dominated by some other allocation. The high-level idea is as follows. We first construct the PILP sentence (which essentially corresponds to the matrix A in Formula (PILP)) assuming that we have a polyhedron Q that describes all envy-free allocations. Then we show how to construct the polyhedron Q such that it meets our assumptions. (In fact, the polyhedron also contains additional technical parts needed to represent that there is an allocation that dominates some allocation from the polyhedron.) Eventually, we use the results from Proposition 2. Starting our proof with assuming that we have polyhedron Q and showing its construction later is due to the fact that the former will develop our intuition how the polyhedron Q should look like. Before we go ahead with the proof, we recall that an allocation $\mathbf{x}$ consists of entries x_i^a, for each agent $a \in \mathcal{A}$ and item type $i \in \mathcal{I}$, with the meaning "we give x_a^i items of item type i to agent a."

Describing Domination of Allocation with Matrix A. Let us assume such a polyhedron Q that we have $(\mathbf{b}, \mathbf{z}) \in Q$, where $\mathbf{z}$ is an allocation (we do not discuss $\mathbf{b}$ as it is still to be defined in the next point of the proof where we construct a proper Q). Our aim is to design a matrix A such that $A\mathbf{x} \leq \mathbf{b}$ if and only if $\mathbf{x}$ is an allocation that dominates $\mathbf{z}$. We first focus on constraints enforcing that $\mathbf{x}$ is a proper allocation (not necessarily allocating all items to the agents; this will be guaranteed later due to the requirement of Pareto-efficiency).

$$\sum_{a \in \mathcal{A}} x_i^a \leq m_i \qquad \forall i \in \mathcal{I} \tag{1}$$

$$x_i^a \geq 0 \qquad \forall a \in \mathcal{A}, \forall i \in \mathcal{I} \tag{2}$$

Condition (1) ensures that $\mathbf{x}$ does not allocate "more items than available," while Condition (2) guarantees that each agent $a \in \mathcal{A}$ is allocated a non-negative number of items by $\mathbf{x}$. It is now not hard to see that $\mathbf{x}$ satisfies Conditions (1) and (2) if and only if $\mathbf{x}$ is a valid allocation.

Thus, it remains to model that $\mathbf{x}$ Pareto-dominates $\mathbf{z}$. One can do so with the following system of inequalities. Note that on the right-hand side we use the (entries of the) vector $\mathbf{z}$; we do so for brevity of our proof. In the final PILP sentence the right-hand side must be defined by $\mathbf{b}$ and we will indeed use the insights from the following inequalities to define $\mathbf{b}$ (as a part of defining Q) in the next step of our proof.

$$\sum_{i \in \mathcal{I}} u_a(i) \cdot x_i^a \geq \sum_{i \in \mathcal{I}} u_a(i) \cdot z_i^a \qquad \forall a \in \mathcal{A} \tag{3}$$

$$\sum_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} u_a(i) \cdot x_i^a \geq 1 + \sum_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} u_a(i) \cdot z_i^a \tag{4}$$

The system of inequalities above guarantees that $\mathbf{x}$ dominates $\mathbf{z}$ if and only if it satisfies Conditions (3) and (4). Note that Condition (3) ensures that the total utility of each agent $a \in \mathcal{A}$ in allocation $\mathbf{x}$ is at least as good as that of agent a in allocation $\mathbf{z}$. Furthermore, given the above, the condition described by Inequality (4) ensures that there is at least one agent $a \in \mathcal{A}$ for whom it holds that $\sum_{i \in \mathcal{I}} u_a(i) \cdot x_i^a > \sum_{i \in \mathcal{I}} u_a(i) \cdot z_i^a$, that is, whose utility is greater in allocation $\mathbf{x}$ than that in $\mathbf{z}$.

The Polyhedron Q. We now aim at designing an appropriate polyhedron Q, existence of which we (only) assumed in the first step. Given the above discussion and Conditions (1)–(4), we have that the claimed $\mathbf{b}$ is in dimension $m+mn+n+1$, that is $\mathbf{b} \in \mathbb{Q}^{m+mn+n+1}$. Indeed, the summands in $\mathbf{b}$'s dimension expression come directly from the numbers of inequalities in, respectively, Conditions (1)–(4). Since we assumed that $\mathbf{z}$ is an allocation, we have $\mathbf{z} \in \mathbb{Z}^{mn}$ by definition. Overall, it must hold that $Q \subseteq \mathbb{Q}^{m+2mn+n+1}$.

Let us now split the vector $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, b_4)$ according to Conditions (1)–(4) above—that is, $\mathbf{b}_1$ is the vector of right-hand sides coming from Condition (1) and so forth. Based on the first two subject conditions, we thus have

$$\mathbf{b}_1 = \mathbf{m}, \tag{5}$$

$$\mathbf{b}_2 = \mathbf{0}, \tag{6}$$

where $\mathbf{m}$ is the vector of item multiplicities. Clearly, if we now use the above-defined $\mathbf{b}_1$ and $\mathbf{b}_2$ substituting the right-hand sides of, respectively, Conditions (1) and (2), the meaning of Conditions (1) and (2) stays intact. More precisely, both conditions still encode the fact that $\mathbf{x}$ is an allocation.

We proceed with constructing vector $\mathbf{b}_3$ and the value of b_4. To achieve this, we first ensure that $\mathbf{z}$ is an envy-free allocation and then derive $\mathbf{b}_3$ and b_4 from this analysis. The following conditions ensure that $\mathbf{z}$ is an envy-free allocation.

$$\sum_{a \in \mathcal{A}} z_i^a \leq m_i \qquad \forall i \in \mathcal{I} \tag{7}$$

$$z_i^a \geq 0 \qquad \forall a \in \mathcal{A}, \forall i \in \mathcal{I} \tag{8}$$

$$\sum_{i \in \mathcal{I}} u_a(i) \cdot z_i^a \geq \sum_{i \in \mathcal{I}} u_a(i) \cdot z_i^{a'} \qquad \forall a, a' \in \mathcal{A} \tag{9}$$

Conditions (7) and (8) ensure that $\mathbf{z}$ encodes an allocation. These expressions and hence the argument are analogous to those of Conditions (1) and (2) for $\mathbf{x}$. Further, Condition (9) ensures that $\mathbf{z}$ is envy-free, since the left-hand side is the total satisfaction of agent a (under allocation $\mathbf{z}$) and the right-hand side is the total value of the bundle of a' viewed via the utility function of agent a (that is, the satisfaction of a if she got the bundle that a' gets under allocation $\mathbf{z}$). At the moment, intuitively, Conditions (5)–(9) describe the "part" of polyhedron Q that defines $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{z}$. What remains, is to define the remaining $\mathbf{b}_3$ and v_4 in a way that we can use them as the right-hand sides of Conditions (4) and (3), respectively. We can do so by binding $\mathbf{z}$ to $(\mathbf{b}_3, b_4)$ as follows, thus obtaining the final two expressions describing polyhedron Q.

$$\sum_{i \in \mathcal{I}} u_a(i) \cdot z_i^a = b_3^a \qquad \forall a \in \mathcal{A} \tag{10}$$

$$\sum_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} u_a(i) \cdot z_i^a = b_4 \tag{11}$$

Observe that the left-hand side of Condition (10) is exactly the right-hand side of (3). Similarly, the right-hand side of (11) contains exactly (up to the constant 1) the right-hand side of Condition (4). Consequently, we can replace the right-hand sides of Conditions (3) and (4) with the right-hand sides of Conditions (10) and (11) while keeping the meaning of the latter unchanged. Observing that in this last step we defined the whole $\mathbf{b}$ in a way that allows us using $\mathbf{b}$ in the right-hand sides of Conditions (1)–(4), we arrive at the next lemma, which summarizes (and follows) from the above discussion.

Lemma 1. *Let $Q \subseteq \mathbb{Q}^{m+2mn+n+1}$ be a polyhedron defined by the conditions* (5)–(11). *Then, $(\mathbf{b}, \mathbf{z}) \in Q$ if and only if*

- $\mathbf{z}$ *is an envy-free allocation of the items described by* $\mathbf{m}$,
- $\mathbf{b}$ *is the vector of right-hand sides of Conditions* (1)–(4).

We remark that the fact that Conditions (9) and (11) are presented in a way that the right-hand side is not a constant is not important in the light of the definition of Q from Lemma 1. Clearly, to obtain a constant on the right-hand sides it is enough to substract the right-hand side from both sides starting from the expressions presented in Conditions (9) and (11).

Using Proposition 2. Having described how to construct the parametric ILP representing EEF–ALLOCATION, we finish the proof of Theorem 1 by applying Proposition 2. More specifically, for a given instance I of the EEF–ALLOCATION problem, we construct matrix A and polyhedron Q as described earlier and directly build a parametric PILP instance I' out of them. Then we run the algorithm from Proposition 2 on instance I'. If the algorithm returns "yes," then for every envy-free allocation there exists one that dominates it, so the answer to the original instance I is "no." In the opposite case, we know that I admits some Pareto-efficient envy-free allocation $\mathbf{x}$, so we output "yes" as an answer to I. Moreover, due to the fact that Proposition 2 guarantees returning a certificate, the "no"-certificate computed by the algorithm is in fact the envy-free Pareto-efficient allocation $\mathbf{x}$.

It remains to analyze the running time of the invocation of the algorithm from Proposition 2 on the constructed instance I'. In the presented model, described by (1)–(11), forming instance I', the dimension of $\mathbf{x}$ is $m \cdot n$, where n is the number of agents in I and m is the number of item types. Hence, the value of parameter p from Proposition 2 is $p = m \cdot n$. It remains to estimate the parameter ϕ thereof. Recall that ϕ is the maximum encoding length of a column in A, which is, in our case, the matrix of left-hand sides in Conditions (1)–(4). The columns of the matrix A are vectors of length $mn + 2m + 1$—this length is equal to the number of constraints (inequalities) required to implement these conditions. Hence, there are $m(n + 2)$ many delimiter symbols in the encoding of a single column. Recall that each such column corresponds to a pair, a single agent $a \in \mathcal{A}$ and a single item $i \in \mathcal{I}$, and let us fix some pair (a, i). So, in the column of (a, i), there are 2 ones, one coming from Condition (1) and one from Condition (2). In addition to this, there are 2 numbers, both equal to $u_a(i)$. Since we assumed a binary encoding, that is $u_a(i) = 2^{o(\log |\mathcal{I}|)}$, we overall obtain the encoding length $2^{o(\log |I|)+1} + m(n+2)$ of a single column, which, after dropping the asymptotically irrelevant terms, gives $\phi = 2^{o(\log |I|)}$. Due to Proposition 1, we thus get that there is a function $\hat{f}(n)$ such that $\phi^{h(n)} \leq \hat{f}(n) \cdot |I|$. Applying this value for $\phi^(h(n))$, together with the one for p shown earlier, proves that the algorithm from Proposition 2 runs in the running time required to show fixed-parameter tractability of EEF–ALLOCATION with respect to the parameter $m + n$.

5 Generalizing Our Approach

Envy-freeness is an appealing yet demanding concept. Consider a very simple example of two agents desiring a single item. Already in this situation an allocation that allocates the item cannot be envy-free. Hence, there is no nontrivial envy-free allocation of items (recall that an empty allocation is always envy-free).

The experimental results of Bredereck et al. [15] give empirical evidence that non-existence of envy-free and Pareto-efficient allocations poses a real threat to applicability of these concepts in real-world instances. The authors show that there were no envy-free and Pareto-efficient allocations for 63% of the instances in their dataset from spliddit.org. The observed phenomenon clearly motivates the need for general approaches. In practice, in the case of a scenario with no envy-free and Pareto-efficient allocation, a reasonable algorithm should not only report the non-existence but also offer a possibly-best alternative allocation, which yields weaker desiderata. The current state of the art in the form of both, an extensive literature on envy-freeness relaxations (see our Related Work section for the references) and general frameworks presented by Bredereck et al. [15, 16] strongly suggest that providing generalizable results is of high value.

Our method meets this criterion and can be used with numerous other problem variants that aim at finding efficient fair allocations. Indeed, it turns out that our technique can be applied to the $\mathcal{E}$-EFFICIENT $\mathcal{F}$-ALLOCATION problem [16], which is a more general variant of the EEF–ALLOCATION where Pareto-efficiency is replaced by some efficiency notion $\mathcal{E}$ and envy-freeness is replaced by some fairness notion $\mathcal{F}$. Formally, the problem, as defined by Bredereck et al. [16], is as follows.

$\mathcal{E}$-EFFICIENT $\mathcal{F}$-ALLOCATION

Input:	A set of agents A, a set of item types I, agent utilities $u_a \colon I \to \mathbb{Z}$ for every $a \in A$, and item multiplicities $m_i \in \mathbb{N}$ for $i \in I$.
Question:	Is there an $\mathcal{F}$-free allocation which is $\mathcal{E}$-efficient.

In fact, our approach can be used to show fixed-parameter tractability of the above problem with respect to the parameterization by the number n of agents plus the number m of item types for various efficiency and fairness notions. Besides relaxed notions of Pareto-efficiency (e.g., where one only cares about being dominated by allocations to some extent similar to the to-be-dominated one) or relaxed envy-freeness such as EF1 [8, 19, 36] or EFX [19, 37], our approach can also deal with generalizations of the concepts of Pareto-optimality such as such as group Pareto-efficiency [2] or generalizations of envy-freeness such as graph envy-freeness [17]. Additionally, our method is adaptable to further somewhat related fairness concepts such as MaxiMinShare [18, 38] or a basic efficiency concept completeness, which only requires that all resources are allocated.

Summarizing, with our technique we can show that $\mathcal{E}$-EFFICIENT $\mathcal{F}$-ALLOCATION is fixed-parameter tractable for parameter $n + m$ even if item multiplicities and utilities are binary encoded when

- $\mathcal{E}$ is a combination of (graph/group) Pareto-efficiency or completeness, and
- $\mathcal{F}$ is a combination of (graph/group) EF, (graph) EF1, (graph) EFX, MaxiMin, or MaxiMinShare.

To avoid repetitiveness, we refer to the work of Bredereck et al. [16] on how to model these notions within the ILP framework.

6 Conclusion

We described a somewhat new usage of Parametric ILPs in fixed dimension in the design of parameterized algorithms, enabling to improve a previous fixed-parameter tractability result. To the best of our knowledge, we are the first to model (and solve) the negation of a given instance to obtain a solution to the original in the context of parameterized complexity. Thus, we believe to have contributed to the, recently gaining increased attention (see, for example, a survey by Gavenčiak et al. [25]), understanding of how the theory of integer (linear) programming impacts the theory of parameterized complexity. We hope our approach leads to further new results in parameterized algorithms, including applications beyond social choice.

Our work also brings up new challenges and highlights the importance of some yet unexplored research directions, mostly in the area of empirical study of efficient and fair allocations of indivisible items.

First of all, given a practically applicable implementation [15] of the approach of Bredereck et al. [16], it appears valuable to pursue an empirical study of our approach as well. It is not uncommon that algorithms with appealing (worst-case) computational complexity guarantees do not perform that well when applied to real-life instances. Hence, designing an implementation of our method and comparing it against the existing methods of computing efficient and fair allocations is a necessary step in judging the usability of our study in practice.

Performing computational experiments is a natural next step to gain additional insights into the problem nature (like, a sharp phase transition in the existence of efficient envy-free allocations reported by Dickerson et al. [21]). Offering a next tool in the algorithmic toolbox for seeking fair allocations, we also highlight the need for further efforts towards obtaining realistic data or, at least, designing diversified synthetic models of generating allocation instances. By now, to the best of our knowledge, except for the relatively small dataset of real-world data from the website spliddit.org [27] and two very simple synthetic models by Dickerson et al. [21], such data is lacking. Our method might not only turn out to be useful in spotting new phenomena of fair allocation instances, but might also well complement other existing methods to form a robust framework for finding fair and efficient allocations.

Acknowledgments

We are thankful to the ECAI'23 reviewers for their helpful comments. This work was started when all authors were with TU Berlin. Dušan Knop acknowledges the support of the Czech Science Foundation Grant No. 22-19557S. Andrzej Kaczmarczyk and Robert Bredereck were supported by the DFG, project AFFA (BR 5207/1 and NI 369/15). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002854).

References

[1] Rediet Abebe, Jon Kleinberg, and David C. Parkes, 'Fair division via social comparison', in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17)*, pp. 281–289, (2017).

[2] Martin Aleksandrov and Toby Walsh, 'Group envy freeness and group pareto efficiency in fair division with indivisible items', in *Proceedings of the 41st German Conference on Artificial Intelligence (KI '18)*, pp. 57–72. Springer, (2018).

[3] Georgios Amanatidis, Georgios Birmpas, and Vangelis Markakis, 'Comparing approximate relaxations of envy-freeness', in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI '18)*, pp. 42–48. AAAI Press, (2018).

[4] Haris Aziz, Sylvain Bouveret, Ioannis Caragiannis, Ira Giagkousi, and Jérôme Lang, 'Knowledge, fairness, and social constraints', in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI '18)*, pp. 4638–4645, (2018).

[5] Haris Aziz, Ioannis Caragiannis, and Ayumi Igarashi, 'Fair allocation of combinations of indivisible goods and chores', *CoRR*, **abs/1807.10684**, (2018).

[6] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh, 'Fair allocation of indivisible goods and chores', in *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19)*, pp. 53–59, (2019).

[7] Wojciech Banaszczyk, Alexander E. Litvak, Alain Pajor, and Stanislaw J. Szarek, 'The flatness theorem for nonsymmetric convex bodies via the local theory of Banach spaces', *Mathematics of Operations Research*, **24**(3), 728–750, (1999).

[8] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish, 'Finding fair and efficient allocations', in *Proceedings of the 19th ACM Conference on Economics and Computation (EC '18)*, pp. 557–574. ACM, (2018).

[9] Xiaohui Bei, Youming Qiao, and Shengyu Zhang, 'Networked fairness in cake cutting', in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI '17)*, pp. 3632–3638. AAAI Press, (2017).

[10] Berhard Bliem, Robert Bredereck, and Rolf Niedermeier, 'Complexity of efficient and envy-free resource allocation: Few agents, resources, or utility levels', in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI '16)*, pp. 102–108. AAAI Press, (2016).

[11] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet, 'Fair allocation of indivisible goods', in *Handbook of Computational Social Choice*, eds., F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, chapter 12, Cambridge University Press, (2016).

[12] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet, *Handbook of Computational Social Choice*, Cambridge University Press, 2016.

[13] Sylvain Bouveret and Jérôme Lang, 'Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity', *Journal of Artificial Intelligence Research*, **32**(1), 525–564, (2008).

[14] Steven J. Brams and Alan D. Taylor, *Fair Division: From Cake-Cutting to Dispute Resolution*, Cambrige University Press, 1996.

[15] Robert Bredereck, Aleksander Figiel, Andrzej Kaczmarczyk, Dušan Knop, and Rolf Niedermeier, 'High-multiplicity fair allocation made more practical', in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '21)*, pp. 260–268, (2021).

[16] Robert Bredereck, Andrzej Kaczmarczyk, Dusan Knop, and Rolf Niedermeier, 'High-multiplicity fair allocation: Lenstra empowered by n-fold integer programming', in *Proceedings of the 2019 ACM Conference on Economics and Computation (EC '19)*, pp. 505–523. ACM, (2019).

[17] Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier, 'Envy-free allocations respecting social networks', *Artificial Intelligence*, **305**, 103664, (2022).

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

[19] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang, 'The unreasonable fairness of maximum nash welfare', in *Proceedings of the 17th ACM Conference on Economics and Computation (EC '16)*, pp. 305–322. ACM, (2016).

[20] Jason Crampton, Gregory Z. Gutin, Martin Koutecký, and Rémi Watrigant, 'Parameterized resiliency problems', *Theoretical Computer Science*, **795**, 478–491, (2019).

[21] John P. Dickerson, Jonathan R. Goldman, Jeremy Karp, Ariel D. Procaccia, and Tuomas Sandholm, 'The computational rise and fall of fairness', in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI '18)*, pp. 1405–1411, (2014).

[22] Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak, 'Parameterized complexity of envy-free resource allocation in social networks', *Artificial Intelligence*, **315**, 103826, (2023).

[23] Friedrich Eisenbrand and Gennady Shmonin, 'Parametric integer programming in fixed dimension', *Mathematics of Operations Research*, **33**(4), 839–850, (2008).

[24] Yotam Gafni, Xin Huang, Ron Lavi, and Inbal Talgam-Cohen, 'Unified fair allocation of goods and chores via copies', *CoRR*, **abs/2109.08671**, (2021).

[25] Tomáš Gavenčiak, Martin Koutecký, and Dušan Knop, 'Integer program-

ming in parameterized complexity: Five miniatures', *Discrete Optimization*, **44**, 100596, (2022).

[26] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica, 'Dominant resource fairness: Fair allocation of multiple resource types', in *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI '11)*, (2011).

[27] Jonathan R Goldman and Ariel D Procaccia, 'Spliddit: Unleashing fair division algorithms', *SIGecom Exchanges*, **13**(2), 41–46, (2014).

[28] Pranay Gorantla, Kunal Marwaha, and Santhoshini Velusamy, 'Fair allocation of a multiset of indivisible items', in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '23)*, pp. 304–331, (2023).

[29] Mark Jones, Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Ondrej Suchý, 'Parameterized complexity of directed Steiner tree on sparse graphs', *SIAM J. Discrete Math.*, **31**(2), 1294–1327, (2017).

[30] Ravi Kannan, 'Test sets for integer programs, $\forall\ \exists$ sentences', in *Proceedings of a DIMACS Workshop on Polyhedral Combinatorics*, eds., William J. Cook and Paul D. Seymour, volume 1 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 39–48. DIMACS/AMS, (1990).

[31] Ravi Kannan, 'Lattice translates of a polytope and the Frobenius problem', *Combinatorica*, **12**(2), 161–177, (1992).

[32] Bart de Keijzer, Sylvain Bouveret, Tomas Klos, and Yingqian Zhang, 'On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences', in *Proceedings of the 1st International Conference on Algorithmic Decision Theory (ADT '09)*, pp. 98–110. Springer, (2009).

[33] Dušan Knop, Martin Koutecký, and Matthias Mnich, 'A unifying framework for manipulation problems', in *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '18)*, pp. 256–264. IFAAMAS, (2018).

[34] Matthias Köppe, Maurice Queyranne, and Christopher Thomas Ryan, 'Parametric integer programming algorithm for bilevel mixed integer programs', *Journal of optimization theory and applications*, **146**(1), 137–150, (2010).

[35] Hendrik W. Lenstra, Jr., 'Integer programming with a fixed number of variables', *Mathematics of Operations Research*, **8**(4), 538–548, (1983).

[36] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi, 'On approximately fair allocations of indivisible goods', in *Proceedings of the 5th ACM Conference on Electronic Commerce (EC '04)*, pp. 125–131. ACM, (2004).

[37] Benjamin Plaut and Tim Roughgarden, 'Almost envy-freeness with general valuations', in *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '18)*, pp. 2584–2603. SIAM, (2018).

[38] Ariel D. Procaccia and Junxing Wang, 'Fair enough: Guaranteeing approximate maximin shares', in *Proceedings of the 15th ACM Conference on Economics and Computation (EC '14)*, pp. 675–692. ACM, (2014).

[39] Jörg Rothe, *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer Texts in Business and Economics, Springer Berlin Heidelberg, 2015.

[40] Larry J. Stockmeyer, 'The polynomial-time hierarchy', *Theoretical Computer Science*, **3**(1), 1–22, (1976).

[41] Toby Walsh, 'Challenges in resource and cost allocation', in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI '15)*, pp. 4073–4077. AAAI Press, (2015).

[42] Toby Walsh, 'Fair division: The computer scientist's perspective', in *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI '20)*, pp. 4966–4972, (2021).

[43] Celia Wrathall, 'Complete sets and the polynomial-time hierarchy', *Theoretical Computer Science*, **3**(1), 23–33, (1976).