Complexity of Efficient and Envy-Free Resource Allocation: Few Agents, Resources, or Utility Levels

Bernhard Bliem
TU Wien, Vienna, Austria
bliem@dbai.tuwien.ac.at

Robert Bredereck and **Rolf Niedermeier**
TU Berlin, Germany
{robert.bredereck,rolf.niedermeier}@tu-berlin.de

Abstract

We study the problem of finding a Pareto-efficient and envy-free allocation of a set of indivisible resources to a set of agents with monotonic preferences, either dichotomous or additive. Motivated by results of Bouveret and Lang [JAIR 2008], we provide a refined computational complexity analysis by studying the influence of three natural parameters: the number n of agents, the number m of resources, and the number z of different numbers occurring in utility-based preferences of the agents. On the negative side, we show that small values for n and z alone do not significantly lower the computational complexity in most cases. On the positive side, devising fixed-parameter algorithms we show that all considered problems are tractable in case of small m. Furthermore, we develop a fixed-parameter algorithm indicating that the problem with additive preferences becomes computationally tractable in case of small n *and* small z.

1 Introduction

Fair allocation of resources is a major theme in society and technology [Walsh, 2015]. In this work, following Bouveret and Lang [2008], we focus on finding fair allocations of indivisible resources (for general information and an overview, refer to the recent survey by Bouveret *et al.* [2016]); herein, fairness translates into "envy-freeness" and (Pareto-)efficiency, two of the most common requirements.

Our general setting is as follows. We are given a set of indivisible resources and a set of agents; each agent may specify preferences in such a way that for any two subsets of the resources we can tell which one (if any) she prefers to the other. The goal is to assign resources to agents such that some fairness criteria are met. We focus on the criteria of being envy-free and efficient. An allocation of resources to agents is called *envy-free* if each agent likes her subset of assigned goods (i.e., her *resource bundle*) at least as much as the bundles assigned to other agents. Since assigning every agent nothing would be envy-free, it is necessary to add a second requirement: efficiency. More precisely, we search for a Pareto-efficient allocation, that is, an allocation of resources where one cannot make any agent better off without making at least one agent worse off. Finally, a few words about the agents' preferences: As Bouveret and Lang [2008] pointed out, allowing for arbitrary preference orders comparing all possible bundles would lead to exponential-size encodings. Hence, more compact representations are needed. We focus on monotonic dichotomous preferences and monotonic additive preferences. The former means that bundles are either acceptable or not, and the acceptable bundles are characterized by monotonic Boolean formulas; the latter means that each bundle has a utility score that is the sum of utilities of the individual resources contained, and each individual utility is non-negative. Refer to Section 1 for formal definitions.

It is not always possible to achieve both Pareto-efficiency and envy-freeness at the same time (e.g., if there are fewer resources than agents and each agent accepts all non-empty bundles). Our central computational problem is as follows.

> EEF-ALLOCATION
> **Input:** A set A of n agents and a set R of m resources. Each agents has preferences over the subsets of R.
> **Question:** Is there an allocation of the resources in R to the agents from A that is Pareto-efficient and envy-free?

Previous results. For both monotonic dichotomous and monotonic additive preferences, EEF-ALLOCATION is known to be complete for Σ_2^P, which is a class on the second level of the Polynomial Hierarchy and consists of all problems that can be solved in nondeterministic polynomial time given an oracle for NP. Bouveret and Lang [2008] also showed intractability (mostly NP-hardness) and polynomial-time solvability of many restricted cases. Further computational complexity studies were performed for the case of incomplete preferences, achieving mostly intractability results [Aziz *et al.*, 2014; Bouveret *et al.*, 2010]. A further slightly different line of research softens the strict requirement of envy-freeness by investigating how to decrease the degree of envy, in this sense taking an "approximate view" on fairness [Lipton *et al.*, 2004; Nguyen and Rothe, 2013]. In summary, considering its high practical relevance, there are comparatively few studies on the computational complexity of EEF-ALLOCATION. In this work, we add to the complexity knowledge by performing a first systematic parameterized complexity analysis for EEF-ALLOCATION.

Our results. We survey our (mostly parameterized) computational complexity results in Table 1. Compared to the focus on classic computational complexity investigations in previous work, we obtain a more fine-grained picture of the complexity landscape. The three parameters we consider are the number n of agents, the number z of numbers for additive preferences (only in combination with n), and the number m of resources. Moreover, for the additive case, we distinguish between unary versus binary encodings of the utility values, and consider the special cases of agents having 0/1 preferences or identical preferences.

A result particularly interesting from a technical point of view is that, using the Boolean Hierarchy, we provide almost tight upper and lower bounds concerning the number of NP oracle calls needed to solve EEF-ALLOCATION with n agents and monotonic dichotomous preferences. Our results further include parameterized hardness results (W[1]-hardness/completeness), and encouraging fixed-parameter tractability (FPT) results, sometimes based on integer linear program formulations exploiting a famous result due to Lenstra [1983] (referred to as ILP-FPT). Since for all parameters even small values may occur in practice (even the case $n = 2$ is frequently studied), our classification results give hope for practically feasible, exact algorithms for the in general Σ_2^P-complete problems. Due to space constraints, we omit many proof details.

Preliminaries on Resource Allocation. We now formalize the notions required for our study of EEF-ALLOCATION.

Definition 1. An *allocation* of a set of resources R to a set of agents A is a mapping $\pi : A \to 2^R$ such that $\pi(a)$ and $\pi(a')$ are disjoint whenever $a \neq a'$. For any agent $a \in A$, we call $\pi(a)$ the *bundle* of a under π. We call π *complete* if $\bigcup_{a \in A} \pi(a) = R$.

In our setting, each agent is associated with a preference relation, which we call the agent's preferences.

Definition 2. A *preference relation* $\preceq$ over a set of resources R is a total preorder over 2^R. We call $\preceq$ *monotonic* if $X \preceq Y$ holds for any $X \subseteq Y \subseteq R$. We write $X \prec Y$ if $X \preceq Y$ but not $Y \preceq X$.

To allow for a succinct representation of preferences, we first consider the restriction that agents either like a bundle or not, according to a Boolean formula over the resources.

Definition 3. We call a preference relation $\preceq$ over a set of resources R *dichotomous* if it is represented by a Boolean formula $\varphi_\preceq$ over R such that for any $X, Y \subseteq R$ it holds that $X \preceq Y$ if and only if $X \models \varphi_\preceq$ implies $Y \models \varphi_\preceq$. We say an allocation π *satisfies* agent a if $\pi(a) \models \varphi_{\preceq_a}$.

For dichotomous preferences, $\preceq$ is monotonic if and only if $\varphi_\preceq$ contains only the connectives $\vee$ and $\wedge$ (in particular no $\neg$).

In the second restriction of preference relations we consider, agents give a utility value to each individual resource. The utility of a bundle is then the sum of the utilities of its elements.

Definition 4. We call a preference relation $\preceq$ over a set of resources R *additive* if there is a *utility function* $u : R \to \mathbb{Z}$ such that for any $X, Y \subseteq R$ it holds that $X \preceq Y$ if and only if $u(X) \leq u(Y)$, where $u(X)$, for $X \subseteq R$, is defined as $\sum_{r \in X} u(r)$.

For additive preferences, $\preceq$ is monotonic if and only if the values of u are non-negative. We call additive preferences *0/1* if each utility function maps to $\{0, 1\}$, and *identical* if each agent has the same utility function.

Given the preferences of each agent, we can define our two desired criteria for allocations.

Definition 5. Let R be a set of resources, let A be a set of agents, and let $\preceq_a$ denote the preference relation of agent $a \in A$. We call an allocation π of R to A *envy-free* if for each pair of agents $a, b \in A$ it holds that $\pi(b) \preceq_a \pi(a)$. We say that an allocation π' of R to A *dominates* another allocation π if for all $a \in A$ it holds that $\pi(a) \preceq_a \pi'(a)$, and for some $a \in A$ it holds that $\pi(a) \prec_a \pi'(a)$. We call π *Pareto-efficient* if there is no allocation of R to A that dominates π, and we call π an *EEF allocation* if it is both Pareto-efficient and envy-free.

Preliminaries on Complexity. To study the parameterized complexity [Cygan *et al.*, 2015; Downey and Fellows, 2013; Flum and Grohe, 2006; Niedermeier, 2006] of EEF-ALLOCATION, we declare some part of the input the *parameter* (e.g., the number of agents or the number of resources). We call a parameterized problem *fixed-parameter tractable* if it is in the class FPTof problems solvable in time $f(\rho) \cdot |I|^{O(1)}$, where $|I|$ is the size of a given instance encoding, ρ is the value of the parameter, and f is an arbitrary computable (typically super-polynomial) function. We use the O^* notation that omits factors polynomial in the input size to emphasize or to compare the super-polynomial factors in the running time. To obtain parameterized intractability, we use a hierarchy of classes of parameterized problems, $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{XP}$. It is widely believed that the first inclusion is proper. The notions of hardness and completeness for parameterized classes are defined through parameterized reductions similar to classical polynomial-time many-one reductions. For this paper, it suffices to additionally ensure that the value of the parameter in the problem we reduce to depends only on the value of the parameter of the problem we reduce from.

The class XP contains all problems that can be solved in $|I|^{O(\rho)}$ time for a function f solely depending on the parameter ρ. Note that containment in XP ensures polynomial-time solvability when ρ is a constant, whereas FPT additionally ensures that the degree of the polynomial is independent of ρ. Unless P = NP, membership in XP can be excluded by showing that the problem is NP-hard for a constant parameter value (for short, we say the problem is para-NP-hard). If the problem is in NP, then we can solve it by one call to an NP oracle (in practice, this means that one can solve the problem with one call of a SAT or Integer Linear Program (ILP) solver).

To show that one oracle call may not suffice, we use many-one reductions to classes BH_n of the Boolean Hierarchy. The class BH_{2k} can be defined as the class of problems Π for which there are problems $P_1, \ldots P_k$, which are in NP, and $Q_1, \ldots, Q_k$, which are in coNP, such that an instance I of Π is a yes-instance if and only if for some i it holds that I is a yes-instance instance of both P_i and Q_i [Wagner, 1987]. To solve BH_n-complete problems, one needs $O(n)$ oracles calls in parallel or $O(\log(n))$ oracle calls in a sequential manner (allowing other computations in between).

preference type	#agents (n)	#agents + #numbers ($n+z$)	#resources (m)
additive 0/1	ILP-FPT (Thm. 5)	ILP-FPT (Thm. 5)	$O^*(m^m/m!)$ (Thm. 1)
additive identical (unary)	W[1]-complete (Thm. 3)	ILP-FPT (Thm. 5)	$O^*(m^m/m!)$ (Thm. 1)
additive (unary)	W[1]-h (Thm. 3), $\in$ XP (Thm. 4)	open	$O^*(m^{2m}/m!)$ (Thm. 1)
additive identical (binary)	para-NP-hard ($n \geq 2$) ♣	ILP-FPT (Thm. 5)	$O^*(m^m/m!)$ (Thm. 1)
additive (binary)	para-NP-hard ($n \geq 2$) ♣	open	$O^*(m^{2m}/m!)$ (Thm. 1)
dichotomous	$\mathsf{BH}_{2^{n-4}}$-hard, $\in \mathsf{BH}_{2^{n+1}}$ (Thm. 2)	n/a	$O^*(m^{2m})$ (Thm. 1)

Table 1: Complexity results for EEF-ALLOCATION with several special cases of monotonic preferences. Results marked by ♣ are from [Bouveret and Lang, 2008].

2 Basic Observations

We start with some observations and introduce some concepts that simplify our investigations in later sections.

For monotonic additive preferences, we assume that for every resource r there is at least one agent a that assigns positive utility to it, that is, $u_a(r) > 0$. Resources that do not satisfy this condition can be safely ignored. We may also assume that no agent assigns utility 0 to all resources because such agents can safely be given nothing as they will not envy anyone and will never be relevant for Pareto-efficiency. Furthermore, we can use an alternative definition of an EEF allocation.

Definition 6. Given an agent set A, a resource set R, and utility functions $u_a : R \to \mathbb{Z}$ encoding the preferences of each agent $a \in A$, an allocation $\pi : A \to 2^R$ is *Pareto-efficient* if

$$\nexists \pi' : \exists a^* \in A : u_{a^*}(\pi'(a^*)) > u_{a^*}(\pi(a^*))$$
$$\wedge \forall a' \in A : u_{a'}(\pi'(a')) \geq u_{a'}(\pi(a')),$$

and *envy-free* if

$$\forall a, a' \in A : u_a(\pi(a)) \geq u_a(\pi(a')).$$

If we further restrict the preferences to be identical or 0/1-preferences, then Pareto-efficiency essentially boils down to completeness. (Note that Bouveret and Lang [2008] use similar observations.) The key insight is that with preferences being identical or 0/1, there is in fact no agent that assigns higher utility to some resource r than the agent to which resource r is allocated. So, by the pigeonhole principle, in order to increase the sum of utilities for one agent, we have to decrease the sum of utilities for another agent.

Observation 1. *In case of monotonic additive identical preferences or 0/1 preferences, an allocation is Pareto-efficient if and only if it is complete and every resource r is allocated to an agent that assigns positive utility to r.*

3 Few Resources

In this section, we discuss the complexity of finding EEF allocations in case of few resources.

Recall from Section 2 that, for monotonic additive preferences, in order to find an EEF allocation, we can safely remove every agent that assigns utility 0 to all resources and every resource to which all agents assign utility 0. Hence, together with a simple preprocessing and brute-force algorithm, we end up with the following. To this end, we say two EEF-ALLOCATION instances are equivalent if they are either both yes-instances or both no-instances.

Proposition 1. *Given an instance of* EEF-ALLOCATION *with monotonic additive preferences and m resources, one can compute an equivalent instance with m resources and at most m agents in linear time. In particular, one can solve* EEF-ALLOCATION *in $O^*(m^{2m})$ time.*

Proof. As we have seen, we may assume that no agent assigns utility 0 to each resource and no resource is assigned utility 0 by all agents. If there are now more than m agents, one will necessarily envy another in every Pareto-efficient allocation. Otherwise we generate each of at most m^m allocations π and check in $O^*(m^m)$ time whether another allocation dominates π to ensure Pareto-efficiency. If π is not dominated, then check in m^2 time whether it is also envy-free. ☐

After preprocessing, both the number n of agents and the number m of resources are linearly bounded in terms of our parameter in case of additive preferences. In contrast, for dichotomous preferences we give an exponential upper bound on the number of agents and show that under reasonable complexity assumptions, no polynomial upper bound exists.

Proposition 2. *Given an instance of* EEF-ALLOCATION *with monotonic dichotomous preferences and m resources, one can compute an equivalent instance with m resources and at most $2^m \cdot (m+1)$ agents in quadratic time. Unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$*, one cannot compute an equivalent instance of size polynomial in m in polynomial time.*

Proof. We assume formulas to be non-tautological (otherwise we could just ignore the respective agents) and to be in CNF or DNF, which allows us to test equivalence by syntactic equality after efficiently removing redundant terms [Mundhenk and Zeranski, 2011]). As the key concept for this proof, we define a maximal set of agents having the same preferences as *agent class*. Observe that there are at most 2^m bundles of resources and an agent can either like a bundle or not. Thus, there are at most $2^{(2^m)}$ agent classes and we can also compute all agent classes in polynomial time (by n^2 tests for equivalence).

Let π be an envy-free allocation and let $A' \subseteq A$ be an agent class. Note that either all agents from A' are satisfied, i.e., $\forall a' \in A' : \pi(a') \models \varphi_{\preceq_{a'}}$ or no agent from A' is satisfied, i.e., $\nexists a' \in A' : \pi(a') \models \varphi_{\preceq_{a'}}$. If $|A'| > m$, only the latter can happen because we cannot satisfy more than number of resources m many agents. To obtain an equivalent instance whose size only depends on a function in m, we exhaustively apply the following simple data reduction rule: If there is some agent class with more than $m+1$ agents, then keep arbitrary $m+1$

of them and remove the rest. In terms of parameterized algorithmics, this instance is a (super-polynomial sized) problem kernel with respect to the number m of resources.

For the kernelization lower bound, we use the *or-cross-composition* framework by Bodlaender *et al.* [2014] with SAT as our source language, using ideas from the proof of Theorem 2. We omit the details due to lack of space. □

By simple brute-force search on this kernel, we obtain fixed-parameter tractability. However, naive brute-force comes with running time $O^*((2^m \cdot (m+1))^{2m})$. Fortunately, by using a smarter algorithm that combines brute-force with the computation of perfect matchings in auxiliary graphs, we can significantly lower the running time. Similar ideas also help improve for additive preferences in case of identical preferences or 0/1 preferences. We summarize our findings as follows.

Theorem 1. EEF-ALLOCATION *with monotonic dichotomous preferences can be solved in* $O^*(B(m)^2)$ *time;* EEF-ALLOCATION *with monotonic additive preferences can be solved in* $O^*(S(m,n)^2 \cdot n!)$ *time;* EEF-ALLOCATION *with additive 0/1 preferences or with monotonic additive identical preferences can be solved in* $O^*(S(m,n))$ *time, where* $B(m) \in O(m^m)$ *denotes the m-th Bell number, and* $S(m,n) \in O(m^m/m!)$ *denotes a Stirling number of the second kind.*

Theorem 1 basically provides FPT-classification results which might be practically relevant for very small values for m. Nevertheless, the corresponding running-time bounds support our impression that more restrictive preference classes simplify the task of finding an EEF allocation.

4 Few Agents

In this section, we discuss the complexity of finding EEF allocations in case of few agents.

Bouveret and Lang [2008] already showed NP-completeness for EEF-ALLOCATION with monotonic dichotomous preferences and only $n = 2$ agents. Although this excludes hope for fixed-parameter algorithms with respect to n, the NP-membership indicates that one might be able to solve EEF-ALLOCATION with only few calls to an NP oracle. Providing almost tight upper and lower bounds, we show how many calls to an NP oracle are needed to solve EEF-ALLOCATION with n agents. This is in line with recent research by de Haan and Szeider; Endriss *et al.* [2014; 2015], which, motivated by the practical success of SAT solvers, investigates whether certain problems can be solved by an algorithm having access to a SAT oracle using FPT time.

Theorem 2. EEF-ALLOCATION *with monotonic dichotomous preferences and* $n \geq 5$ *agents is in* $\mathsf{BH}_{2^{n+1}}$ *and* $\mathsf{BH}_{2^{n-4}}$*-hard.*

Proof. Membership. Note that the fact that each instance has n agents allows us to assume that the set A of agents is the same over all instances (albeit the preferences may differ). Let $A_1, \ldots, A_{2^n}$ denote all subsets of A. Now, for any $1 \leq i \leq 2^n$, let P_i be the problem of deciding whether there is an envy-free allocation satisfying exactly the agents in A_i, and let Q_i be the problem of deciding whether there is no allocation $\pi : A \to 2^R$ satisfying all agents in A_i and one additional agent.

P_i is in NP: We can clearly check in polynomial time if an allocation satisfies exactly the agents in A_i and is envy-free.

The complement of Q_i is in NP: We can check in polynomial time if an allocation satisfies A_i and an additional agent.

Obviously there is an EEF allocation for an instance if and only if this instance is a yes-instance for some P_i (indicating that an envy-free allocation π satisfies exactly A_i) and at the same time for Q_i (indicating that no allocation dominates π).

Hardness (construction only). We set $n' := 2^n - 5$ and provide a polynomial-time reduction from the following $\mathsf{BH}_{2n'}$-hard problem [Wagner, 1987]: Given Boolean formulas $\chi_1, \ldots, \chi_{n'}$ and $\psi_1, \ldots, \psi_{n'}$, we ask whether for some i it holds that χ_i is satisfiable while ψ_i is unsatisfiable. We assume w.l.o.g. that the variables of any pair of formulas are disjoint and formulas are non-empty. We also assume that each formula is in negation normal form, i.e., no binary connectives other than $\vee$ and $\wedge$ occur, and negation occurs only directly in front of variables. Given any set S, we write S' to denote $\{v' \mid v \in S\}$. For any Boolean formula α, we denote the set of variables in α by $\mathrm{Var}(\alpha)$, and for any i we define $V_i = \mathrm{Var}(\chi_i) \cup \mathrm{Var}(\psi_i)$ as well as $L_i = V_i \cup V_i'$. Let χ_i^* and ψ_i^* be the result of replacing each literal of the form $\neg v$ by v' in χ_i and ψ_i, respectively.

In the following, we construct an EEF-ALLOCATION instance with agents $1, \ldots, n$ having as preferences $\varphi_1, \ldots, \varphi_n$, respectively. Let the set of resources be $\{x, y\} \cup L_1 \cup \cdots \cup L_{n'}$.

$$\begin{aligned}
\varphi_1 &= x, \\
\varphi_2 &= x \vee \bigwedge_{v \in V_1} (v \vee v') \vee \cdots \vee \bigwedge_{v \in V_{n'}} (v \vee v'), \\
\varphi_3 &= x \vee \chi_1^* \vee \cdots \vee \chi_{n'}^*, \\
\varphi_4 &= y \wedge (\psi_1^* \vee \cdots \vee \psi_{n'}^*), \\
\varphi_5 &= \varphi_4.
\end{aligned}$$

The intuition behind the remaining $n - 5$ agents is that after satisfying them, only the literals for at most one pair of formulas χ_i^*, ψ_i^* are left for distribution. If $n' = 4$, for example, we realize this by means of an additional agent that claims $L_1 \cup L_2$ or $L_3 \cup L_4$, and another agent that claims L_1, L_2, L_3, or L_4. This can be generalized such that one agent claims one half of all literals, the next claims one quarter, and so on. Formally, for $1 \leq i \leq n-5$ and $0 \leq j \leq 2^i - 1$, we define $P_{i,j} = L_{j \cdot n'/2^i + 1} \cup \cdots \cup L_{j \cdot n'/2^i + n'/2^i}$. For $1 \leq i \leq n-5$, we now define $\varphi_{5+i} = x \vee \bigvee_{j=0}^{2^i-1} \bigwedge_{l \in P_{i,j}} l$. This reduction is doable in polynomial time since n is logarithmic in n'. The resulting EEF-ALLOCATION instance is a yes-instance if and only if some χ_i is satisfiable while ψ_i is unsatisfiable. Due to lack of space, we only give some intuition: Agent 1 must receive x in any EEF allocation π, which forces π to satisfy all agents except 4 and 5. Agent 3's bundle encodes a model for some χ_i, as the agents $6, \ldots, n-5$ and 2 prevent her from getting both v and v' for any variable v. If ψ_i were satisfiable, there would be envy between agents 4 and 5 under π since π is efficient. □

This result entails that the parameter n does not allow us to reduce the problem to, e.g., SAT in FPT time, which would have been good news due to the Σ_2^P-completeness of the general problem. In fact, membership in $\mathsf{BH}_{2^{n+1}}$ enables us to solve an EEF-ALLOCATION instance with 2^{n+1} *parallel* NP oracle calls or $O(n)$ *sequential* NP oracle calls. If we are not only interested in solving the decision problem but, e.g., also in computing a solution, there are good (rather technical complexity-theoretic) reasons to believe that the number of sequential oracle calls may further increase [Jenner and Torán, 1995]. Actually, $O(n^2m)$ sequential NP oracle calls are sufficient for finding an EEF-allocation: Consider the extended version of EEF-ALLOCATION where the input additionally specifies a partial allocation of some resources to agents and one asks for an EEF-allocation that extends the partial allocation. Start with the empty partial allocation and extend it step by step to a complete EEF-allocation as follows. Iteratively take an unallocated resource r_j and find an agent to which r_j can be allocated in an EEF-allocation by solving the extended EEF-ALLOCATION problem (using $O(n)$ sequential NP oracle calls each time). To see that the extended EEF-ALLOCATION problem can indeed be solved with only $O(n)$ sequential NP oracle calls, one can use a $\mathsf{BH}_{2^{n+1}}$-membership proof that works analogously to the proof of Theorem 2.

We next consider the case of additive preferences, for which Bouveret and Lang [2008] have shown NP-hardness even in the case of identical preferences and $n = 2$ agents. However, their hardness result relies on preferences encoded in binary, which leaves the possibility open that a pseudo-polynomial-time algorithm exists. We show that the problem with preferences encoded in unary is W[1]-complete if all agents have the same preferences, which implies NP-hardness and W[1]-hardness for the parameter n if the agents may have different preferences. In Theorem 4, we will show that the latter W[1]-hardness can be complemented by showing that the problem can be solved in polynomial time for constant n.

Theorem 3. EEF-ALLOCATION *with monotonic additive preferences encoded in unary is* W[1]*-complete with respect to the number n of agents if all agents have the same preferences.*

Proof. For W[1]-hardness, we give a parameterized reduction from the UNARY BIN PACKING problem, which is W[1]-hard with respect to the number b of bins [Jansen *et al.*, 2013].

> UNARY BIN PACKING
> **Input:** Positive integers $w_1, \ldots, w_m, b, C$ in unary.
> **Question:** Is there an assignment of m items with weights $w_1, \ldots, w_m$ to at most b bins such that none of the bins exceeds weight capacity C?

The set of resources consists of a set $R := \{r_1, \ldots, r_m\}$ of *item resources* and a set $D := \{d_1, \ldots, d_q\}$ of *dummy resources*, where $q := b \cdot C - \sum_{1 \le i \le m} w_i$ is the total amount of "unused capacity". Create agents $a_1, \ldots, a_b$ such that each agent a_i has a utility function u_i that sets $u_i(d_j) = 1$ for any dummy resource d_j and $u_i(r_j) = w_j$ for each item resource r_j.

We show that there is an assignment of m items with weights $w_1, \ldots, w_m$ to at most b bins such that none of the bins exceeds capacity C if and only if there is an EEF allocation.

"Only if" direction: Assume that there is an assignment of m items with weights $w_1, \ldots, w_m$ to at most b bins such that none of the bins exceeds capacity C. For each bin j, let $P(j)$ denote the set of items assigned to j and let $W(j) := \sum_{i \in P(j)} w_i$ denote the total weight of the items assigned to j. We create an EEF allocation π as follows. For each $1 \le j \le b$, $\pi(j)$ contains all item resources corresponding to the items from $P(j)$ and further $C - W(j)$ dummy resources. First, no agent envies any other agent since each receives a bundle with utility sum C. Second, π is also Pareto-efficient by Observation 1, since π is complete and all utilities are positive.

"If" direction: Assume that there is an EEF allocation π. Recall that π must be complete due to Observation 1, so the total sum of utilities over all bundles in π is exactly $b \cdot C$. Since π is Pareto-efficient and envy-free, the total utility $\sum_{r_j \in \pi(i)} u_i(r_j)$ of the bundle allocated to agent i is exactly C: It is the same for each agent because the agents have the same preferences and the allocation is envy-free. Hence, assigning all items corresponding to the resource items of bundle $\pi(i)$ to bin i is a solution for the UNARY BIN PACKING instance. This completes the W[1]-hardness proof. For W[1]-membership we can reuse ideas from the hardness proof. ☐

5 Few agents and Few Utility Levels

In this section, we focus on instances of EEF-ALLOCATION with monotonic additive preferences where the utility functions of the agents have only few different values. To this end, let z denote the number of different values that occur in the utility functions (c.f. [Fellows *et al.*, 2012]), and let $z_{\max}$ denote the maximum number that occurs in any utility function. Since EEF-ALLOCATION is NP-hard already for additive 0/1 preferences ($z = 2$ and $z_{\max} = 1$) [Bouveret and Lang, 2008], we further assume that there are only few agents to obtain positive results.

The decisive advantage of monotonic additive preferences is that knowing all $u_a(\pi(a'))$ values for all allocations π and all pairs (a, a') of agents is enough to check the existence of an EEF allocation. By a dynamic programming algorithm based on this observation, we show that EEF-ALLOCATION with monotonic additive preferences encoded in unary can be solved in polynomial time if the number n of agents is a constant. Note that in case of preferences encoded in unary, z and $z_{\max}$ are upper-bounded by the instance size.

Theorem 4. EEF-ALLOCATION *with monotonic additive preferences can be solved in* $O((m \cdot z_{\max} + 1)^{n^2} \cdot mn^2)$ *time.*

Proof. We denote the agents by integers $1, \ldots, n$ and use the following $(n^2 + 1)$-dimensional binary table where entry

$$T(k, x_{1,1}, \ldots, x_{1,n}, x_{2,1}, \ldots, x_{2,n}, x_{n,1}, \ldots, x_{n,n})$$

(for $1 \le k \le m$ and $0 \le x_{i,j} \le m \cdot z_{\max}$) is 1 if and only if there is an allocation π that (i) assigns each of the first k resources to some agent and (ii) ensures that $u_i(\pi(j)) = x_{i,j}$ for $1 \le i, j \le n$.

After computing the table T our algorithm answers "yes" if there is some entry $T(m,\ x_{1,1}, \ldots, x_{1,n},\ x_{2,1}, \ldots, x_{2,n},\ \ldots,\ x_{n,1},\ \ldots, x_{n,n}) = 1$ such that the following hold: (i) For all $i, j \in \{1, \ldots, n\}$ it holds that $x_{i,i} \ge x_{i,j}$; and (ii) there is no entry $T(m,\ x'_{1,1}, \ldots, x'_{1,n},\ x'_{2,1}, \ldots, x'_{2,n},\ \ldots,\ x'_{n,1},\ \ldots, x'_{n,n}) = 1$ where $\exists 1 \le i^* \le n : x'_{i^*,i^*} > x_{i^*,i^*}$ and $\forall 1 \le i \le n : x'_{i,i} \ge x_{i,i}$; otherwise, the algorithm answers "no". We now show how to compute this table in polynomial time.

In the following, for technical reasons we assume that $T(k,\ x_{1,1}, \ldots, x_{1,n},\ x_{2,1}, \ldots, x_{2,n},\ x_{n,1}, \ldots, x_{n,n}) = 0$ whenever one tries to access an entry out of the table's range (e.g., with $k < 1$ or $x_{i,j} < 0$). We compute all entries $T(k, x_{1,1}, \ldots, x_{1,n}, x_{2,1}, \ldots, x_{2,n}, x_{n,1}, \ldots, x_{n,n})$ assuming that all entries $T(k-1, x_{1,1}, \ldots, x_{1,n}, x_{2,1}, \ldots, x_{2,n}, x_{n,1}, \ldots, x_{n,n})$ are known. The key insight is that for each allocation π' of the first k resources, there is an allocation π of the first $k-1$ resources such that resource r_k can be assigned to some agent i^* and the following equations hold for every agent i: (a) $u_i(\pi'(i^*)) = u_i(\pi(i^*)) + u_i(r_k)$, and (b) $u_i(\pi'(j)) = u_i(\pi(j))$ for $j \neq i^*$. Formally, we set the entry $T(k, x_{1,1}, \ldots, x_{1,n}, x_{2,1}, \ldots, x_{2,n}, \ldots, x_{n,1}, \ldots, x_{n,n})$ to 1 if and only if there is some $i^* \in \{1, \ldots, n\}$ such that

$$T(k-1, y_{1,1}, \ldots, y_{1,n}, \ldots, y_{n,1}, \ldots, y_{n,n}) = 1,$$

where, for any $1 \leq i, j \leq n$, we define $y_{i,j}$ as

$$y_{i,j} = \begin{cases} x_{i,j} - u_i(r_k) & \text{if } j = i^*, \\ x_{i,j} & \text{otherwise.} \end{cases}$$

For correctness, note that if our algorithm answers "yes", then there clearly is an EEF allocation by the definition of the table T. Now, assume towards a contradiction that the algorithm answers "no", although there is an EEF allocation π. This means that either entry $T(m, u_1(\pi(1)), \ldots, u_1(\pi(n)), u_2(\pi(1)), \ldots, u_2(\pi(n)), u_n(\pi(1)), \ldots, u_n(\pi(n))) = 0$ or this entry is 1 but there is an entry $T(m, x'_{1,1}, \ldots, x'_{1,n}, x'_{2,1}, \ldots, x'_{2,n}, \ldots, x'_{n,1}, \ldots, x'_{n,n}) = 1$ where $\exists 1 \leq i^* \leq n : x'_{i^*,i^*} > u_{i^*}(\pi(i^*))$ and $\forall 1 \leq i' \leq n : x'_{i',i'} \geq u_{i'}(\pi(i'))$. The former would imply that π does not allocate all m resources, which cannot be true since π is Pareto-efficient and we assume that for each resource r, there is at least one agent a with $u_a(r) > 0$. The latter would imply that π is not Pareto-efficient—a contradiction.

As for the running time, observe that the table is of size $m \cdot (m \cdot z_{\max} + 1)^{n^2}$, which is a polynomial in the instance size if the number n of agents is a constant and $z_{\max}$ is encoded in unary. To compute a table entry, one has to check at most n previous table entries (involving n subtractions per check). □

The algorithm from Theorem 4 can be used not only to decide the problem but also to actually find an EEF allocation. To this end, one basically replaces the binary entries by concrete allocations.

Parameterized by n and z alone, EEF-ALLOCATION is para-NP-hard: Bouveret and Lang [2008] showed NP-hardness for EEF-ALLOCATION with two agents and for EEF-ALLOCATION with 0/1 preferences. Theorem 4 gives XP membership for parameter n and unary encoding of the utility values. Since the latter implies that z and $z_{\max}$ are polynomially upper-bounded by the instance size, cases where n and z are both small seem to be easier to solve. We further explore this by considering the combined parameter $(n+z)$ and the parameter n for 0/1 preferences.

Theorem 5. EEF-ALLOCATION *is fixed-parameter tractable (i) with respect to the combined parameter $(n+z)$ for identical monotonic additive preferences, and (ii) with respect to n alone for additive 0/1 preferences, where n denotes the number of agents and z is the number of different values that occur in the utility functions.*

Proof. We define the *fingerprint* of some resource r_j to be the size-n integer vector $f_j := (u_1(r_j), u_2(r_j), \ldots, u_n(r_j))$. We denote the set of fingerprints by F and the number of resources that have fingerprint f by $\#(f)$. Observe that there are at most z^n different fingerprints and that, given an EEF allocation π, exchanging any two resources with the same fingerprint between two bundles in π results in another EEF allocation. Moreover, we can characterize EEF allocations by solely specifying how many resources with fingerprint f are allocated to agent a_i for each agent $a_i \in A$ and each fingerprint f.

Using the fingerprint concept, we develop an integer linear programming formulation as follows. For each agent $a_i \in A$ and each fingerprint f, create one integer variable x_i^f that denotes the number of resources with fingerprint f allocated to agent a_i.

We achieve envy-freeness by adding the constraints

$$\forall 1 \leq i \leq n,\ 1 \leq i' \leq n : \sum_{f \in F} x_i^f \cdot f[i] \geq \sum_{f \in F} x_{i'}^f \cdot f[i], \qquad (1)$$

where $f[i]$ denotes the i-th entry in fingerprint f.

To obtain Pareto-efficiency, we add the constraint sets

$$\forall f \in F : \sum_{1 \leq i \leq n} x_i^f = \#(f), \text{ and} \qquad (2)$$

$$\forall f \in F,\ 1 \leq i \leq n \text{ with } f[i] = 0 : x_i^f = 0. \qquad (3)$$

Constraint set (2) ensures completeness and Constraint set (3) ensures that each resource is allocated to an agent that assigns positive utility to this resource. Hence, due to Observation 1, we obtain Pareto-efficiency.

Finally, fixed-parameter tractability follows by the famous result of Lenstra [1983] (later improved by Kannan; Frank and Tardos [1987; 1987]) that says that an ILP with ρ variables and ℓ input bits can be solved in $O(\rho^{2.5\rho+o(\rho)}\ell)$ time. □

6 Conclusion

The goal of this work was to contribute to a more fine-grained understanding of the computational complexity landscape of EEF-ALLOCATION for monotonic preferences (dichotomous and additive), which is Σ_2^P-complete in general. On the positive side, we obtained encouraging fixed-parameter tractability results, indicating practical feasibility of relevant special cases. Indeed, it remains to explore whether the efficiency of the algorithms can be further improved—our focus was on complexity classification rather than engineering best possible running times—and, in the same spirit, to consider even more special but practically relevant cases such as a fixed small number of agents. Indeed, in the long run implementation and experiments with the developed algorithms seems promising. To this end, also studying further natural parameterizations could be helpful. A different theoretical route would be to extend the investigations also to incomplete preferences [Aziz *et al.*, 2014; Bouveret *et al.*, 2010] or approximate envy-freeness [Lipton *et al.*, 2004; Nguyen and Rothe, 2013].

Acknowledgments This work was supported by the Austrian Science Fund (FWF), project Y698, and by the German Research Foundation (DFG), project AFFA (NI 369/15, BR 5207/1).

References

[Aziz *et al.*, 2014] Haris Aziz, Serge Gaspers, Simon Mackenzie, and Toby Walsh. Fair assignment of indivisible objects under ordinal preferences. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '14)*, pages 1305–1312, 2014.

[Bodlaender *et al.*, 2014] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics*, 28(1):277–305, 2014.

[Bouveret and Lang, 2008] 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.

[Bouveret *et al.*, 2010] Sylvain Bouveret, Ulle Endriss, and Jérôme Lang. Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In *Proceeding of the 19th European Conference on Artificial Intelligence (ECAI '10)*, pages 387–392. IOS Press, 2010.

[Bouveret *et al.*, 2016] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 12. Cambridge University Press, 2016.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Downey and Fellows, 2013] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

[Endriss *et al.*, 2015] Ulle Endriss, Ronald de Haan, and Stefan Szeider. Parameterized complexity results for agenda safety in judgment aggregation. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*, pages 127–136. IFAAMAS, 2015.

[Fellows *et al.*, 2012] Michael R. Fellows, Serge Gaspers, and Frances A. Rosamond. Parameterizing by the number of numbers. *Theory of Computing Systems*, 50(4):675–693, 2012.

[Flum and Grohe, 2006] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[Frank and Tardos, 1987] András Frank and Éva Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

[de Haan and Szeider, 2014] Ronald de Haan and Stefan Szeider. Fixed-parameter tractable reductions to SAT. In *Proceeding of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *LNCS*, pages 85–102. Springer, 2014.

[Jansen *et al.*, 2013] Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39–49, 2013.

[Jenner and Torán, 1995] Birgit Jenner and Jacobo Torán. Computing functions with parallel queries to NP. *Theoretical Computer Science*, 141(1&2):175–193, 1995.

[Kannan, 1987] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

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

[Lipton *et al.*, 2004] 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)*, pages 125–131, 2004.

[Mundhenk and Zeranski, 2011] Martin Mundhenk and Robert Zeranski. How to apply SAT-solving for the equivalence test of monotone normal forms. In *Proceedings of the 40th International Conference on Theory and Applications of Satisfiability Testing (SAT '11)*, volume 6695 of *LNCS*, pages 105–119. Springer, 2011.

[Nguyen and Rothe, 2013] TrungThanh Nguyen and Jörg Rothe. How to decrease the degree of envy in allocations of indivisible goods. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT '13)*, volume 8176 of *LNCS*, pages 271–284. Springer, 2013.

[Niedermeier, 2006] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[Wagner, 1987] Klaus W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1–2):53–80, 1987.

[Walsh, 2015] Toby Walsh. Challenges in resource and cost allocation. In *Proceedings of the 29sth AAAI Conference on Artificial Intelligence*, pages 4073–4077. AAAI Press, 2015.