Several Stories about High-Multiplicity EFx Allocation (Student Abstract)

Nikita Morozov[1], **Artur Ignatiev**[2], **Yuriy Dementiev**[2]

[1]Constructor University, 28759 Bremen, Germany
[2]HSE University, St. Petersburg 190121, Russia
nmorozov@jacobs-university.de, aaignatiev@hse.ru, ydementyev@hse.ru

Abstract

Fair division is a topic that has significant social and industrial value. In this work, we study allocations that simultaneously satisfy definitions of fairness and efficiency: EFx and PO. First, we prove that the problem of finding such allocations is NP-hard for two agents. Then, we propose a concept for an ILP-based solving algorithm, the running time of which depends on the number of EFx allocations. We generate input data and analyze algorithm's running time based on the results obtained.

Introduction and Related Work

Fair and efficient allocation of resources is a very important issue in Economics and Computer Science. First mentioned in the mid-20th century, it arises in a variety of practical applications, such as dividing rewards among groups, allocating students to courses, and assigning tasks within a team.

One of the most popular notion of fairness is envy-freeness (EF), which requires that every agent prefers their own bundle of goods to that of any other. However in the case of indivisible goods, EF allocations may not exist. This motivated the study of its relaxations. One of the actual and relevant relaxations of EF proposed by (Caragiannis et al. 2016) is called envy-free up to any item (EFx). Each agent's bundle should be worth at least as much as any other agent's bundle minus any single item for the allocation to be EFx. The existence of EFx allocations is considered as the biggest open question in fair division. We refer to overview paper (Amanatidis et al. 2023) for a detailed overview of fair division of indivisible goods.

The standard notion of efficiency is Pareto optimality (PO). An allocation is said to be PO if no other one makes an agent better off without making someone else worse off.

An important question in fair division is whether the notions of fairness can be achieved in conjunction with the efficiency notions PO. In general, $\mathrm{EFx} + \mathrm{PO}$ allocations are not guaranteed to exist (Plaut and Roughgarden 2020). In this paper we focus on the algorithmic complexity of finding $\mathrm{EFx} + \mathrm{PO}$ (EFx and at the same time PO allocation).

Setting. There is a set N of n agents and a set M of m goods that cannot be divided or shared. Each agent $i \in N$ is equipped with an additive valuation function $v_i : 2^M \to \mathbb{N}_{\geq 0}$, which assigns a non-negative integer and $v_i(S) = \sum_{g \in S} v_i(g)$ for any subset of items $S \subseteq M$. A fair allocation instance is denoted by $I = (N, M, v)$ where $v = (v_1, ..., v_n)$ is the vector of valuation functions and can be represented by a table with a row per agent and a column per good, such that cell (i, g) contains the value $v_i(g)$. An allocation is a tuple of subsets of $M : A = (A_1, \dots, A_n)$, such that each agent $i \in N$ receives the bundle, $A_i \subseteq M, A_i \cap A_j = \emptyset$ for every pair of agents $i, j \in N$, and $\bigcup_{i \in [n]} A_i = M$.

Definition 1. *An allocation* A *is envy-free up to any item (*EFx*) if it satisfies:*

$$\forall i, j \in N : \; u_i(A_i) + \min_{z \in A_j} u_i(z) \geq u_i(A_j).$$

Definition 2. *An allocation* A *is pareto-optimal (*PO*) if there is no other allocation B such that:*

$$\begin{cases} \forall\, i \in N : \; u_i(B_i) \geq u_i(A_i), \\ \exists\, j \in N : \; u_j(B_j) > u_j(A_j). \end{cases}$$

Item type is a vector of length n, where the i-th coordinate is the value of the good's utility for the i-th agent. We will use k to denote the number of item types. In other words, there are exactly k unique goods. The introduction of item types immediately entails the typification of allocations. We do not distinguish allocations that differ from each other by permutation items of the same type. That is, by the number of allocations we mean the number of different (up to permutation of items of the same type) allocations.

Hardness

On the one hand, the $\mathrm{EFx} + \mathrm{PO}$ problem is NP-hard for 3-valued instances (Garg and Murhekar 2021), so it rules out the existence of an algorithm solving $\mathrm{EFx} + \mathrm{PO}$ in FPT time ($f(k') \cdot |I|^{O(1)}$ for parameter k' and some function f) with the parameter number of values (unless $\mathsf{P} = \mathsf{NP}$). On the other hand, we show that $\mathrm{EFx} + \mathrm{PO}$ problem is NP-hard for two agents.

Theorem 1 (in Appendix)**.** *The problem of existence of an* $\mathrm{EFx} + \mathrm{PO}$ *allocation is* NP*-hard.*

In (Bredereck et al. 2020) authors show (theoretical) fixed-parameter tractability results for finding envy-free and pareto-optimal allocation with parameter n agents and k types. We are concentrating on a more practical algorithm for EFx + PO problem possible with an additional parameter and explanation of its efficiency.

Integer Linear Programming

For simplicity of presentation, we will formulate an ILP for two agents. General case can be found in the appendix (our approach is similar to (Bredereck et al. 2021)).

EFx restrictions for two agents can be simplified as follows:

$$0 \le x_i \le m_i,\ 0 \le i \le k. \tag{1}$$

$$\sum_{i=1}^{k} a_i x_i + \min_{j:\, m_j - x_j \neq 0} a_j \ge \sum_{i=1}^{k} a_i (m_i - x_i). \tag{2}$$

$$\sum_{i=1}^{k} b_i (m_i - x_i) + \min_{j:\, x_j \neq 0} b_j \ge \sum_{i=1}^{k} b_i x_i. \tag{3}$$

Here x_i is the variable meaning number of goods of type i that the first agent has, a_i, b_i—the utility of the object of type i for the first and the second agents, m_i—the number of goods of type i.

The minimum condition can be replaced with a number of if-statements and binary variables, which is shown in the appendix along with PO condition.

Our algorithm consists of three parts: we find an EFx allocation, check it for PO and then repeat it if the allocation is not EFx + PO. A few words have to be said about the last step. It is important to search only for the allocations that were not analyzed yet. In order to do so initial ILP instance must be replenished with restrictions.

At the end we have an ILP with at most $p = (2s+1)kn + 4kn^2$ variables where s is the number of EFx allocations. We have to solve it not more than s times, so the overall runtime is $O^*(p^{2.5p+o(p)})$ (Lenstra 1983).

Number of EFx allocations

Our key goal is to evaluate the number of the EFx allocations as it appears in the runtime estimation of ILP (runtime of a similar ILP for finding EF and PO allocation can be found in (Bredereck et al. 2021)). We implemented the brute force method to count the number of such allocations.

We work on a problem with 2 agents to reduce the computational costs. It also allowed us to find the upper bound on the number of EFx allocations. The proof that the number of allocations does not exceed $(\lceil \frac{m+k}{k} \rceil)^k$ is provided in the appendix. An example of an instance where the number of EFx allocations is $\Omega(\frac{m^k}{2^k k^k})$ can also be found there.

Generating inputs. We decided to implement our own input generator. It takes n, m, k and maximal weight as parameters and returns an instance. An instance is chosen from the uniform distribution over all the instances with k types and bounded by maximal weight value weights.

After these theoretical results we decided to run our algorithm on the smaller instances. It was made to show that

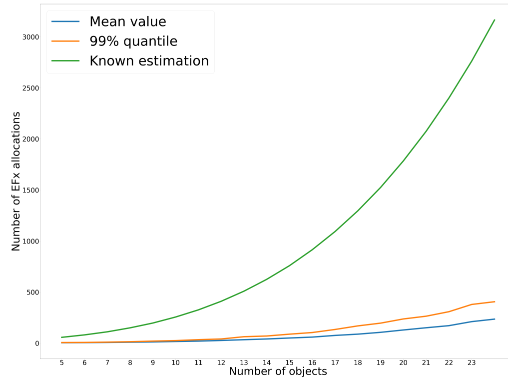

Figure 1: EFx allocation number, max weight=10, k = 4.

even though the upper bound is high the estimated number of allocations is significantly lower. The results are shown on the Fig.1. Even the 0.99 quantile is much lower than the upper bound. Because of that we expect our algorithm to work even faster than the estimated runtime. Experiments with various metrics (such as the percentage of EFx allocations that are PO) can be found in supplemental material.

Results and Discussions. We see that the number of EFx allocations is not so large in practice. But is it possible to prove some probabilistic upper bound on the number of EFx allocations? Also indirectly the complexity of such algorithms is affected by the percentage of EFx allocations that are PO. Is it possible to prove some lower bound on this percentage?

Acknowledgments

Artur Ignatiev and Yuriy Dementiev: This work was done under support of the grant №075-15-2022-289 for creation and development of Euler International Mathematical Institute. This work was supported by HSE University (Basic Research Program).

References

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

Bredereck, R.; Figiel, A.; Kaczmarczyk, A.; Knop, D.; and Niedermeier, R. 2021. High-Multiplicity Fair Allocation Made More Practical. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, 260–268. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

Bredereck, R.; Kaczmarczyk, A.; Knop, D.; and Niedermeier, R. 2020. High-Multiplicity Fair Allocation Using Parametric Integer Linear Programming. arXiv:2005.04907.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2016. The Unreasonable Fairness of Maximum Nash Welfare. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, 305–322. New York, NY, USA: Association for Computing Machinery. ISBN 9781450339360.

Garg, J.; and Murhekar, A. 2021. Computing Fair and Efficient Allocations with Few Utility Values. In *Algorithmic Game Theory: 14th International Symposium, SAGT 2021, Aarhus, Denmark, September 21–24, 2021, Proceedings*, 345–359. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-85946-6.

Lenstra, H. W. 1983. Integer Programming with a Fixed Number of Variables. *Math. Oper. Res.*, 8(4): 538–548.

Plaut, B.; and Roughgarden, T. 2020. Almost Envy-Freeness with General Valuations. *SIAM J. Discret. Math.*, 34(2): 1039–1068.