

Comparative study in fair division algorithms

Liad Nagi and Moriya Elgrabli

Instructor: Erel Segal Halevi

June 8, 2022

Abstract

A comparison of four fair division algorithms performed on real data from the spliddit website. The comparison was made on the sum of agent's utilities, and the minimum utility for an agent in an allocation.

1 Introduction

A fair division algorithm is an algorithm that divides a set of resources among several people who have an entitlement to them so that each person receives their due share. The algorithm takes into account the utilities of the items to each person. This paper compares fair division algorithms in minimum utility and the sum of all agent's utilities.

2 Method

We implemented the three-quarters MMS allocation algorithm from the "An Improved Approximation Algorithm for Maximin Share" article by Jugal Garg and Setareh Taki [GT21], into the fairpy open-source library in python ¹. We compared the performance of this algorithm to three other fair division algorithms implemented in fairpy: Max sum allocation, leximin [Wil98], and PROPm [BGGS21].

We conducted a comparison of 730 cases of item divisions. The data was collected from the spliddit website [GP15] and was kindly shared with us by Nisarg Shah. The data contains 730 cases of items and their evaluations by various agents.

We compared the algorithms in two ways:

- The minimum utility an agent receives
- The sum of the utilities of all agents

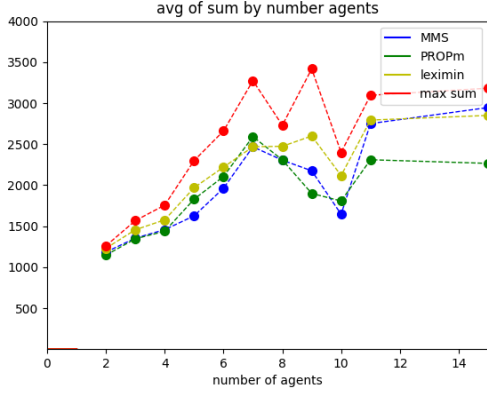
The comparison was made as an average of the results of each number of agents separately. The average amount of utilities was calculated for one agent, two agents, and so on. Similarly, the minimum utility of an agent in an allocation was calculated separately for each number of agents.

Since the three-quarters MMS allocation algorithm stops when each agent gets at least three-quarters of his MMS value, it does not divide all items. This can cause a significant gap in performance measurement, so in order to solve this, we added a division of the remaining items as follows: start from the last agent and give him an item he values as more than 0 (if any) and continue dividing the items to other agents until the items run out. We started from the last agent because the Three-quarters MMS allocation algorithm takes care of the first agents first ².

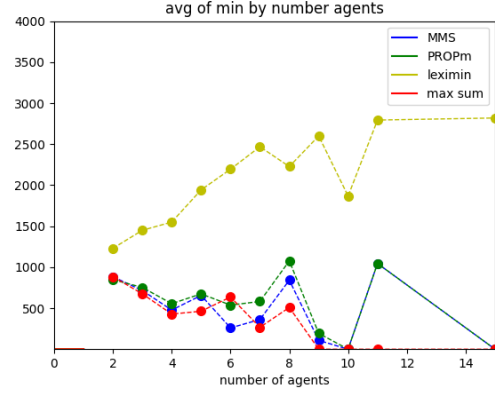
¹https://github.com/erelsgl/fairpy/blob/master/fairpy/items/approximation_maximin_share.py

²https://github.com/erelsgl/fairpy/blob/master/experiments/compare_algorithms.py

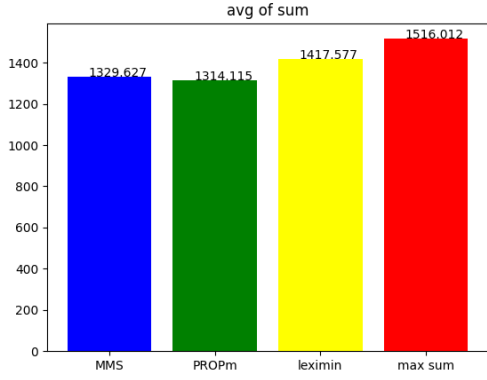
3 Comparative study



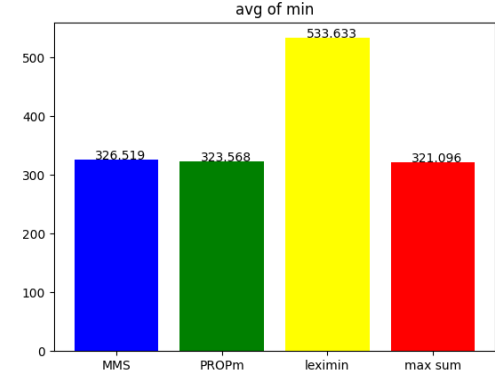
(a) The average sum of agents utilities, for each amount of agents



(b) The average of the minimum utility for an agent per allocation, for each amount of agents



(c) The average sum of agents utilities



(d) The average of the minimum utility for an agent per allocation

4 Results

As might be expected, for the sum of utilities, the algorithm that maximizes the sum of the utilities returns a significantly better result, and the rest of the algorithms are close in terms of performance. In the same way, for the minimum utilities, leximin is significantly better in terms of performance - which makes sense, since leximin allows the division of items in the middle, and therefore the final utility each agent receives can be higher (e.g. if there are two people interested in only one item, both will receive a certain utility), while if items are distributed in their entirety - the minimum utility will be significantly harmed.

5 Conclusions

PROPm and the three-quarters MMS allocation algorithms are close in performance. Although both are based on different calculation methods and try to ensure different fairness conditions, in the end, both return very similar results.

6 Acknowledgments

We would like to thank Jugal Garg and Setareh Taki, the authors of the article we implemented, who helped us understanding the article. In addition, we would like to thank Nisarg Shah who shared with us and allowed us to use real data from the spliddit website to perform the experiments on.

References

- [BGGS21] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. Propm allocations of indivisible goods to multiple agents. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 24–30. International Joint Conferences on Artificial Intelligence Organization, 8 2021.
- [GP15] Jonathan Goldman and Ariel D Procaccia. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges*, 13(2):41–46, 2015.
- [GT21] Jugal Garg and Setareh Taki. An improved approximation algorithm for maximin shares. *Artificial Intelligence*, 300(103547), 2021.
- [Wil98] Stephen J Wilson. Fair division using linear programming. *preprint, Departement of Mathematics, Iowa State University*, 1998.