Lecture Note 8: Fair Allocation with Indivisible Goods

Christian Kroer*

February 22, 2022

1 Introduction

In this lecture note we study the problem of performing fair allocation when the items are indivisible. This setting presents a number of challenges that were not present in the divisible case.

It is obviously an important setting in practice. For example, the website `http://www.spliddit.org/` allows users to fairly split estates, financial assets, toys, or other goods. Another important application is that of fairly allocating course seats to students. This setting is even more intricate, because valuations in that setting are combinatorial. In order to design suitable mechanisms for fairly dividing discrete goods, we will need to reevaluate our fairness concepts.

2 Setup

We have a set of m indivisible goods that we wish to divide among n agents. We assume that each good has supply 1. We will denote the bundle of goods given to agent i as x_i, where x_{ij} is the amount of good j that is allocated to buyer i. The set of feasible allocations is then $\{x | \sum_i x_{ij} \leq 1, x_{ij} \in \{0,1\}\}$

Unless otherwise specified, each agent is assumed to have a linear utility function $u_i(x_i) = \langle v_i, x_i \rangle$ denoting how much they like the bundle x_i.

3 Fair Allocation

In the case of indivisible items, several of our fairness properties become much harder to achieve. We will assume that we are required to construct a Pareto-efficient allocation.

Proportional fairness doesn't even make sense anymore: it rested on the idea of assigning each agent their fractional share $\frac{1}{n}$ of each item. There is however, a suitable generalization of proportionality that does make sense for the indivisible case: the *maximin share (MMS) guarantee*: For agent i, their MMS is the value they would get if they get to divide the items up into n bundles, and are required to take the worst bundle. Formally:

$$\begin{aligned} \max_{x \geq 0} \ & \min_j u_i(x_j) \\ \text{s.t.} & \sum_i x_{ij} \leq 1, \forall j \\ & x_{ij} \in \{0,1\}, \forall i, j \end{aligned}$$

*Department of Industrial Engineering and Operations Research, Columbia University. Email: `christian.kroer@columbia.edu`.

We say that an allocation x is an MMS allocation if every agent i receives utility $u_i(x_i)$ that is at least as high as their MMS guarantee. In the case of 2 agents, an MMS allocation always exists. As an exercise, you might try to come up with an algorithm for finding such an allocation[1]

In the case of 3 or more agents, such a solution may not exist. The counterexample is very involved, so we won't cover it here.

Theorem 1. *For $n \geq 3$ agents, there exist additive valuations for which an MMS allocation does not exist. However, an allocation such that each agent receives at least $\frac{3}{4}$ of their MMS guarantee always exists.*

The original spliddit algorithm for dividing goods worked as follows: first, compute the $\alpha \in [0, 1]$ such that every agent can be guaranteed an α fraction of their MMS guaranteee (this always ends up being $\alpha = 1$ in practice). Then, subject to the constraints $u_i(x_i) \geq \alpha \text{MMS}_i$, a social welfare-maximizing allocation was computed. However, this can lead to some weird results.

Example 1. *Three agents each have valuation 1 for 5 items. In that case, the MMS guarantee is 1 for each agent. But now the social welfare-maximizing solution can allocate three items to agent 1, and 1 item each to agents 2 and 3. Obviously a more fair solution would be to allocate 2 items to 2 agents, 1 item to the last agent.*

One observation we can make about the 3/1/1 solution versus the 2/2/1 solution is that envy is strictly higher in the 3/1/1/ solution.

With the above motivation, let us consider envy in the discrete setting. It is easy to see that we generally won't be able to get envy-free solutions if we are required to assign all items. Consider 2 agents splitting an inheritance: a house worth \$500k, a car worth \$10k, and a jewelry set worth \$5k. Since we have to give the house to a single agent, the other agent is guaranteed to have envy. Thus we will need a relaxed notion of envy:

Definition 1. *An allocation x is* envy-free up to one good *(EF1) if for every pair of agents i, k, there exists an item j such that $x_{kj} = 1$ and $u_i(x_i) \geq u_i(x_k - e_k)$, where e_k is the k'th basis vector.*

Intuitively, this definition says that for any pair of agents i, k such that i envies k, that envy can be removed by removing a single item from the bundle of k. Note that requiring EF1 would have forced us to use the 2/2/1 allocation in Example 1.

For linear utilities, an EF1 allocation is easily found (if we disregard Pareto optimality). As an exercise, come up with an algorithm for computing an EF1 allocation for linear valuations[2] In fact, EF1 allocations can be computed in polynomial time for any monotone set of utility functions (meaning that if $x_i \geq x_i'$ then $u_i(x_i) \geq u_i(x_i')$).

However, ideally we would like to come up with an algorithm that gives us EF1 as well as Pareto efficiency. To achieve this, we will consider the product of utilities, which we saw previously in Eisenberg-Gale. This product is also called the *Nash welfare* of an allocation:

$$NW(x) = \prod_i u_i(x_i)$$

[1]Solution: compute one of the solutions solutions to agent 1's MMS computation problem. Then let agent 2 choose their favorite bundle, and give the other bundle to agent 1. Agent 1 clearly receives their MMS guarantee, or better. Agent 2 also does: their MMS guarantee is at most $\frac{1}{2}\|v_2\|_1$, and here they receive utility of at least $\frac{1}{2}\|v_2\|_1$.

[2]This is achieved by the round-robin algorithm: simply have agents take turns picking their favorite item. It is easy to see that EF1 is an invariant of the partial allocations resulting from this process.

The *max Nash welfare* (MNW) solution picks an allocation that maximizes $NW(x)$:

$$\max_x \prod_i u_i(x_i)$$
$$s.t. \sum_i x_{ij} \leq 1, \forall j$$
$$x_{ij} \in \{0,1\}, \forall i,j$$

Note that here we have to worry about the degenerate case where $NW(x) = 0$ for *all* x, meaning that it is impossible to give strictly positive utility to all agents. We will assume that there exists x such that $NW(x) > 0$. If this does not hold, typically one seeks a solution that maximizes the number of agents with strictly positive utility, and then the largest MNW achievable among subsets of that size is chosen.

The MNW solution turns out to achieve both Pareto optimality (obviously, since otherwise it would not solve the MNW optimization problem), and EF1:

Theorem 2. *The MNW solution for linear utilities is Pareto optimal and EF1.*

Proof. Let x be the MNW solution. Say for contradiction that agent i envies agent k by more than one good. Let j be the item allocated to agent k that minimizes the ratio $\frac{v_{kj}}{v_{ij}}$. Let x' be the same allocation as x, except that $x'_{ij} = 1, x'_{kj} = 0$. The proof is concluded by showing that $NW(x') > NW(x)$, which contradicts optimality of x for the MNW problem.

Using the linearity of utilities we have $u_i(x'_i) = u_i(x_i) + v_{ij}$ and $u_k(x'_k) = u_k(x_k) - v_{kj}$. Every other utility stays the same. Now we have

$$\begin{aligned} &\frac{NW(x')}{NW(x)} > 1 \\ \Leftrightarrow &\frac{[u_i(x_i) + v_{ij}] \cdot [u_k(x_k) - v_{kj}]}{u_i(x_i) u_k(x_k)} > 1 \\ \Leftrightarrow &\left[1 + \frac{v_{ij}}{u_i(x_i)}\right] \cdot \left[1 - \frac{v_{kj}}{u_k(x_k)}\right] > 1 \\ \Leftrightarrow &\frac{v_{kj}}{v_{ij}} [u_i(x_i) + v_{ij}] < u_k(x_k) \end{aligned} \tag{1}$$

By how we choose j we have

$$\frac{v_{kj}}{v_{ij}} \leq \frac{\sum_{j' \in x_k} v_{kj'}}{\sum_{j' \in x_k} v_{ij'}} \leq \frac{u_k(x_k)}{u_i(x_k)},$$

and by the envy property we have

$$u_i(x_i) + v_{ij} < u_i(x_k).$$

Now we can multiply together the last two inequalities to get (1). □

The MNW solution also turns out to give a guarantee on MNW, but not a very strong one: every agent is guaranteed to get $\frac{2}{1+\sqrt{4n-3}}$ of their MMS guarantee, and this bound is tight. Luckily, in practice the MNW solution seems to fare much better. On Spliddit data, the following ratios are achieved. In the table below are shown the MMS approximation ratios across 1281 "divide goods" instances submitted to the Spliddit website for fairly allocating goods

MMS approximation ratio intervals	$[0.75, 0.8)$	$[0.8, 0.9)$	$[0.9, 1)$	1
% of instances in interval	0.16%	0.7%	3.51%	95.63%

Over 95% of the instances have every player receive their full MMS guarantee.

4 Computing Discrete Max Nash Welfare

4.1 Complexity

The problem of maximizing Nash welfare is generally not easy. In fact, the problem turns out to be not only NP-hard, but NP-hard to approximate within a factor $\mu \approx 1.00008$ (the best currently-known approximation factor is 1.45, so the gap between 1.00008 and 1.45 is open).

The reduction is based the vertex-cover problem on 3-regular graphs, which is NP-hard to approximate within factor ≈ 1.01. A 3-regular graph is a graph where each vertex has degree 3.

The proof is not particularly illuminating, so we will skip it here. However, let's see a quick way to prove a simpler statement: that the problem is NP-hard even for 2 players with identical linear valuations. Consider the following

Definition 2. Partition *problem: you are given a multiset of integers $S = \{s_1, \ldots, s_m\}$ (potentially with duplicates), and your task is to figure out if there is a way to partition S into two sets S_1, S_2 such that $\sum_{i \in S_1} s_i = \sum_{i \in S_2} s_2$.*

We may now construct an MNW instance as follows: we create two agents and m items. Each agent has value s_j for item j. Now by the AM-GM inequality (2d case: $\sqrt{xy} \leq \frac{x+y}{2}$, with equality iff $x = y$) there exists a correct partitioning if and only if the MNW allocation has value $(\frac{1}{2} \sum_j s_j)^2$.

This result can be extended to show strong NP-hardness by considering the k-equal-sum-subset problem: given a multiset $\mathcal{S}$ of $x_1, \ldots, x_n$ positive integers, are there k nonempty disjoint subsets $S_1, \ldots, S_k \subset \mathcal{S}$ such that $sum(S_1) = \ldots = sum(S_k)$. The exact same reduction as before works, but with k agents rather than 2.

4.2 Algorithms

Given these computational complexity problems, how should we compute an MNW allocation in practice?

We present two approaches here. First, we can take the log of the objective, to get a concave function. After taking logs, we get the following mixed-integer exponential-cone program:

$$
\begin{aligned}
\max \quad & \sum_i \log u_i & & \\
s.t. \quad & u_i & \leq \langle v_i, x_i \rangle, \quad & \forall i = 1, \ldots, n \\
& \sum_i x_{ij} & \leq 1, \quad & \forall j = 1, \ldots, m \\
& x_{ij} & \in \{0, 1\}, \quad & \forall i, j
\end{aligned}
\tag{2}
$$

This is simply the discrete version of the Eisenberg-Gale convex program. One approach is to solve this problem directly, e.g. using Mosek.

Alternatively, we can impose some additional structure on the valuation space: if we assume that all valuations are integer-valued, then we know that $u_i(x_i)$ will take on some integer value in the range 0 to $\|v_i\|_1$. In that case, we can add a variable w_i for each agent i, and use either (1) the linearization of the log at each integer value, or (2) the linear function from the line segment $(\log k, k), (\log(k+1), k+1)$, as upper bounds on w_i. This gives $\frac{1}{2}\|v_i\|_1$ constraints for each i using the line segment approach (the linearization uses twice as many constraints), but ensures that w_i is equal to $\log\langle v_i, x_i \rangle$ for all integer-valued $\langle v_i, x_i \rangle$. Using the line segment approach gives the following mixed-integer linear program (MILP):

$$\begin{aligned}
\max \quad & \sum_i w_i \\
s.t. \quad & w_i \leq \log k + [\log(k+1) - \log k] \times (\langle v_i, x_i \rangle - k), && \forall i = 1, \ldots, n, k = 1, 3, \ldots, \|v_i\|_1 \\
& \sum_j v_{ij} x_{ij} \geq 1, && \forall i \\
& \sum_i x_{ij} \leq 1, && \forall j = 1, \ldots, m \\
& x_{ij} \in \{0, 1\}, && \forall i, j
\end{aligned} \tag{3}$$

These two mixed-integer programs both have some drawbacks: For the first mixed-integer exponential-cone program, we must resort to much less mature technology than for mixed-integer linear programs. On the other hand, the discrete EG program is reasonably compact: the program is roughly the size of a solution. For the MILP, the good news is that MILP technology is quite mature, and so we might expect this to solve quickly. On the other hand, adding $n \times \|v_i\|_1$ additional constraints can be quite a lot, and could lead to slow LP solves as part of the branch-and-bound procedure.

Figure 1 shows the performance of the two approaches.

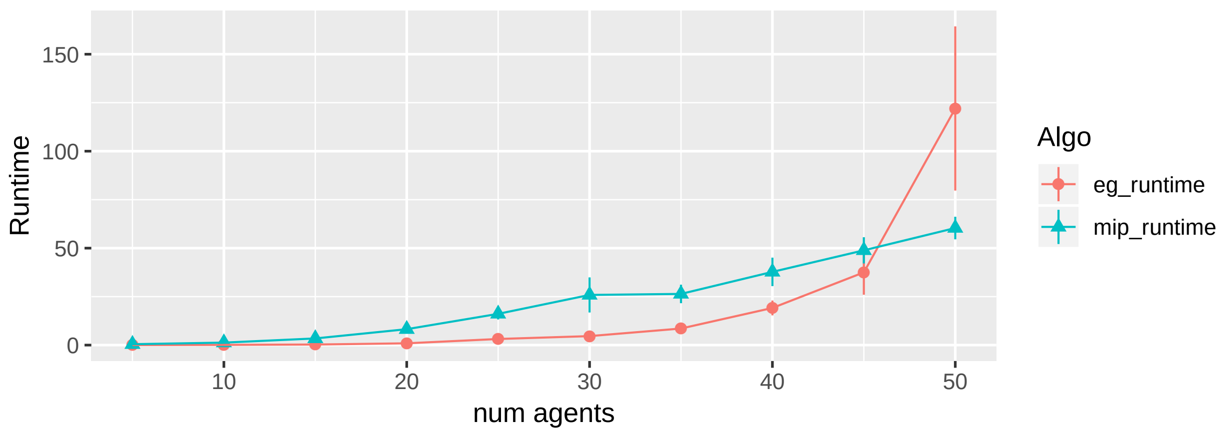

Figure 1: Plot showing the runtime of discrete Eisenberg-Gale and the MILP approach.

5 Historical Notes

The maximin share was introduced by Budish [2]. The results on nonexistence of MMS allocation, and an approximation guarantee of $\frac{2}{3}$ were given by Kurokawa et al. [7]. The approximation guarantee was improved to $\frac{3}{4}$ by Ghodsi et al. [6]. The application of MNW to fair division was proposed by Caragiannis et al. [3].

A really nice overview talk targeted at a technical audience is given by Ariel Procaccia here: `https://www.youtube.com/watch?v=7lUtS-l9ytI`. Most of the material here is based on his excellent presentations of these topics.

The 1.00008 inapproximability result was by Lee [8]. The 1.45-approximation algorithm was given byBarman et al. [1]. Strong NP-hardness of k-EQUAL-SUM-SUBSET is shown in Cieliebak et al. [5].

The MILP using approximation to the log at each integer point was introduced by Caragiannis et al. [4]. At the time, Mosek did not support exponential cones, and so they did not compare to the direct solving of discrete Eisenberg-Gale. The results shown here are the first direct comparison of the two, as far as I know.

References

[1] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 557–574, 2018.

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

[3] 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 2016 ACM Conference on Economics and Computation*, pages 305–322. ACM, 2016.

[4] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.

[5] Mark Cieliebak, Stephan J Eidenbenz, Aris Pagourtzis, and Konrad Schlude. On the complexity of variations of equal sum subsets. *Nord. J. Comput.*, 14(3):151–172, 2008.

[6] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 539–556, 2018.

[7] David Kurokawa, Ariel D Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM (JACM)*, 65(2):1–27, 2018.

[8] Euiwoong Lee. Apx-hardness of maximizing nash social welfare with indivisible items. *Information Processing Letters*, 122:17–20, 2017.