IEOR8100: Economics, AI, and Optimization
Lecture Note 15: Fair Division with Indivisible Goods

Christian Kroer*

May 4, 2020

1 Introduction

In this lecture note we study the fair division setting where 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. 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. Without loss of generality 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 Division

In the case of indivisible items, several of our fairness properties become much harder to achieve. We will assume that we are require 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,

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

and are required to take the worst bundle. Formally:

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

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 $u_i(x_i) \geq u_i(x'_i)$).

[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.

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:

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

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

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

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

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

By how we choose j we have

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

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):

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

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.

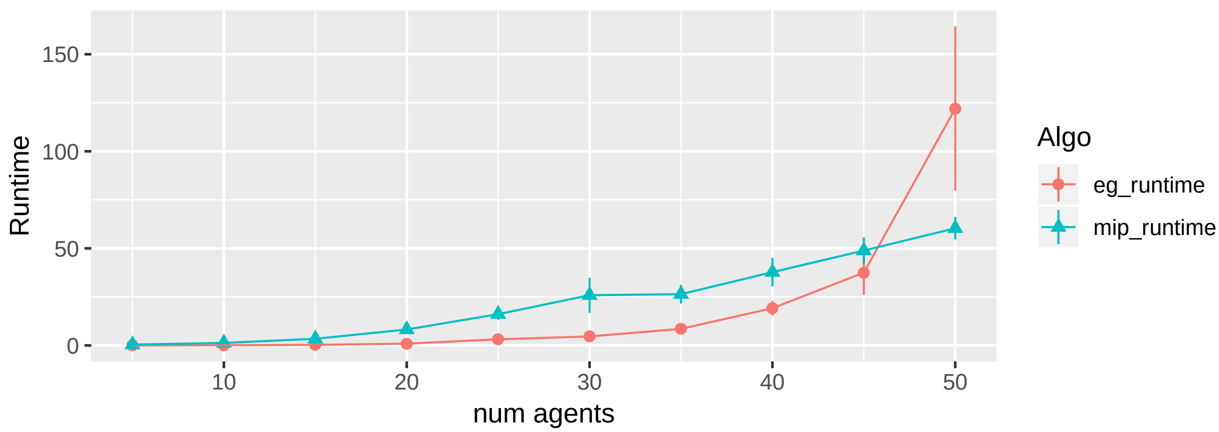

Figure 1: Plot showing the runtime of discrete Eisenberg-Gale and the MILP approach.

5 Market Equilibrium

We saw previously that in the divisible case *competitive equilibrium from equal incomes* (CEEI) is a very good method for achieving a fair allocation. Motivated by this, we may ask whether CEEI could also be used to achieve fair outcomes for the indivisible case? However, even for the case of linear utilities, we quickly hit a hurdle: since we cannot achieve envy-free solutions, there is no way that we can guarantee the existence of a market equilibrium when all buyers have the same budget (why?).

As mentioned in a prior note, a *market equilibrium* is a set of prices $p \in \mathbb{R}_+^m$ for each of the m goods, as well as an allocation $x \in \{0,1\}^{n \times m}$ of goods to buyers such that everybody is assigned an optimal allocation given the prices and their budget. Formally, the *demand set* of an buyer i with budget B_i is

$$D(p) = \text{argmax}_{x_i \geq 0} u_i(x_i) \text{ s.t. } \langle p, x_i \rangle \leq B_i$$

A market equilibrium is an allocation-price pair (x, p) s.t. $x_i \in D(p)$ for all buyers i, and $\sum_i x_{ij} = 1$.

5.1 Approximate CEEI

Since a market equilibrium is not guaranteed to exist for equal budgets (or for many other budget allocations), we will instead look at *approximate CEEI* (A-CEEI). In A-CEEI the idea is to relax two parts of CEEI: (1) we give agents approximately equal, rather than exactly equal, budgets, and (2) we only clear the market approximately.

Let's see how this works with an example. Consider an example where two agents are trying to divide four goods: two diamonds (one large (LD), one small (SD)), and two rocks (one pretty (PR), one ugly (UR)). Say the agents both have utilities such that they can take at most two items, and they prefer bundles in the order

$$(LD, SD) > (LD, PR) > (LD, UR) > (LD) > (SD, PR) > (SD, UR) > (PR, UR) > (PR) > (UR).$$

Clearly if budgets are equal we cannot hope to price these items in a way that clears the market, since both agents will always want the bundle with the large diamond if they can afford it. But if we instead give agent 1 a budget of 1.2 and agent 2 a budget of 1, then we can set the prices as follows:

LD	SD	PR	UR
1.10	0.8	0.2	0.1

Now agent 1 wishes to buy (LD, UR) for a total price of 1.2, and agent 2 wishes to buy (SD, PR) for a total price of 1. As long as we decide the budget perturbations in a randomized way this is in some sense fair in expectation, and furthermore we might hope that the budget perturbations are small enough that for instances with more than four items, things look even fairer. Note that the allocation we found satisfies both EF1 and the MMS guarantee. The example also achieves Pareto optimality, but we will in general only guarantee approximate Pareto optimality for A-CEEI for more general valuations.

We will describe the problem in the context of matching students to seats in courses. This setup is used in the *Course Match* software, which is used for matching students at Wharton and several other schools. There is a set of m courses, and each course j has some capacity s_j. There is a set of n students. Each student has a set $\Psi_i \subseteq 2^m$ of feasible subsets of courses that they may be allocated, with each bundle containing at most $k \leq m$ courses (note that this assumes that each student can only consume one unit of a good, even if $s_j > 1$; this is of course reasonable in course allocation, but not for all applications). The set Ψ_i encodes both scheduling constraints such as courses meeting at the same time, as well as constraints specific to the student such as whether they satisfy the prerequisites. The preferences of student i are assumed to be given as a complete and transitive ordinal preference ordering $\succcurlyeq_i$ over Ψ_i. Completeness simply means that for all schedules $x, x' \in \Psi_i$, $x \succcurlyeq_i x'$, $x' \succcurlyeq_i x$, or both. Transitivity means that if $x \succcurlyeq_i x'$ and $x' \succcurlyeq_i x''$ then $x \succcurlyeq_i x''$.

Given a set of prices p for each course, a vector x_i^* is in the demand set for student i if

$$x_i^* \in \text{argmax}_{\succcurlyeq_i} \{x_i \in \Psi_i : \langle x_i, p \rangle \leq B_i\}.$$

In the actual Course Match implementation, $\succcurlyeq_i$ is represented numerically by an utility function for each student, but the A-CEEI theory works for the more general case of ordinal preferences.

Since we have existence issues (these arise both from indivisibility as seen earlier, but also from the very general preference orderings allowed), we resort to an approximation to CEEI:

Definition 3. *An allocation x, prices p, and budgets B constitute an (α, β)-CEEI if:*

1. $x_i \in \operatorname{argmax}_{\succcurlyeq} \{x' \in \Psi_i : \langle p, x' \rangle \leq B_i\}$ *for all i*
2. $\|z\|_2 \leq \alpha$*, where $z \in \mathbb{R}^m_+$ is defined as $z_j = \sum_i x_{ij} - s_j$ if $p_j > 0$, and $z_j = \max(\sum_i x_{ij} - s_j, 0)$ if $p_j = 0$*
3. $B_i \in [1, 1+\beta]$ *for all i*

The first condition in (α, β)-CEEI simply says that each student i buys an item in their demand set. The second condition says that supply constraints are approximately satisfied. The third constraint says that all budgets are almost the same, up to a difference of β.

The main theorem regarding (α, β)-CEEI is that they are guaranteed to exist:

Theorem 3. *Let $\sigma = \min(2k, m)$. For any $\beta > 0$, there exists a $(\sqrt{\sigma m}/2, \beta)$-CEEI. Moreover, given budgets $B \in [1, 1+\beta]^n$ and any $\epsilon > 0$, there exists a $(\sqrt{\sigma m}/2, \beta)$-CEEI using budgets B^* such that $\|B^* - B\|_\infty \leq \epsilon$.*

One major concern with this result is that we are not quite guaranteed a feasible solution. In general the allocation may oversubscribe some courses, though the oversubsciption vector z has bounded ℓ_2 norm. In practice, the bound is relatively modest: First, the bound $\sqrt{\sigma m}/2$ does not grow with the number of agents or number of course seats. Second, in practice students take at most a modest number of courses per semester among a reasonably-small number of courses offered (an example given in the literature is that students take $k = 5$ courses out of 50 courses total at Harvard's MBA program), thus yielding a bound of roughly 11. Technically a single course could be oversubscribed by 11 students, but in practice we expect this to be smoothed out reasonably across many courses.

The proof of the existence theorem is rather involved and relies on smoothing out the market in order to invoke fixed-point theorems. Here we give some intuition for the role that each approximation plays.

As in other discontinuous settings, the main difficult for existence without approximation is the discontinuity of student demands with respect to price. However, in the course match setting, $\sqrt{\sigma}$ is an upper bound on the discontinuity of the demand of any single agent. To see this, note that a demand x_i has at most k entries set to 1, and so a student can at most drop all courses from x_i and switch to k new courses under their new demand x'_i. At the same time, there's only m courses total, so the change is bounded by $\min(2k, m)$, and thus $\|x_i - x'_i\|_2 \leq \sqrt{\sigma}$.

The second discontinuity issue is to avoid large discontinuous aggregate changes in demand across the students. When budgets are the same, as in standard CEEI, the demand discontinuity across students may occur at the same point in the space of prices. Thus, if this happens, aggregate discontinuity may be on the order of $n\sigma$. With distinct budgets, it becomes possible to change a single student's demand without changing those of other students. For each bundle x, we may think of the hyperplane $H(i, x) = \{p : \langle p, x \rangle \leq B_i\}$ which denotes the boundary between two halfspaces in the price space: those where student i can afford x, and those where i cannot afford x. By having each budget distinct, one can show that in a generic sense, at most m hyperplanes can intersect at any particular point in price space. This implies that aggregate demand changes by at most σm.

The remainder of the proof is concerned with smoothing out the aggregate demands so that a fixed-point existence theorem can be applied to show existence.

5.2 Fairness and Optimality Properties of A-CEEI

Since we are only approximately clearing the market, we do not get Pareto optimality. However, it is possible to show that if we construct a modified market where $\tilde{s}_j = s_j - z_j$, then we have Pareto optimality in that market. Thus, any Pareto-improving allocation must utilize unused supply, which can potentially be used to bound the inefficiency once more structure is imposed on utilities.

Crucially, (α, β)-CEEI does guarantee some fairness properties. If we select $\beta \leq \frac{1}{k-1}$, then EF1 is guaranteed in any (α, β)-CEEI. Furthermore, there exists β small enough such that each student is also guaranteed to receive their $(n+1)$-MMS share, which is their utility if they were forced to partition the items into $n+1$ bundles and take the worst one.

5.3 Practical Course Match Concerns

In Course Match, the representation of $\succcurlyeq_i$ is as follows: the set of feasible schedules Ψ_i is taken as given. Then, student i ranks each course on a scale from $0 - 100$, and is additionally allowed to specify pairwise penalties or bonuses in $-200, 200$ for being assigned a given pair of courses.

5.4 Computing A-CEEI

In general computing an A-CEEIis PPAD complete. This is the same class of problem that general-sum Nash equilibrium falls in. It is conjectured to require exponential time in the worst case, and thus we cannot hope to have nice scalable algorithms like we had for the divisible case.

In practice, A-CEEI is computed using local search. A *tabu search* is used on the space of prices. This works as follows:

1. A price vector is generated randomly
2. A set of "neighbors" are generated using two different generation approaches:
   - "Price gradient:" all the demands under the current prices are added up, and the excess demand vector is treated as a gradient. Then, 20 different stepsizes are tried along the price gradient
   - A single item has its price changed, and all other prices are kept the same. The new price on the chosen item is set high enough to stop it from being oversubscribed, or low enough to stop being underscribed. A neighbor is generated for each over or undersubscribed item
3. The best neighbor (among the ones generating a previously-unseen allocation) is selected as the next price vector, and the procedure repeats from step 2 (unless the last 5 iterations yielded no improving prices, in which case the local search stops)
4. Finally, step 1 is repeated with a new random price vector. This repeats until a time limit is reached

In practice this procedure generates an A-CEEI solution with significantly better α and β values than the theory predicts, within roughly two days of computation. In the process, about 4.25 billion MIPs are solved. After an A-CEEI has been generated, additional heuristics are implemented in order to force the solution to not have oversubscription.

6 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. [4].

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.

A-CEEI was introduced by Budish [2], and an implementation of A-CEEI used at Wharton was given by Budish et al. [3].

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] Eric Budish, Gérard P Cachon, Judd B Kessler, and Abraham Othman. Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, 65(2):314–336, 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 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.