Computing Competitive Equilibria with Mixed Manna*

Jugal Garg
University of Illinois at Urbana Champaign
jugal@illinois.edu

Peter McGlaughlin
University of Illinois at Urbana Champaign
mcglghl2@illinois.edu

ABSTRACT

Fair division is the problem of allocating a set of items among a set of agents in a fair and efficient way. It arises naturally in a wide range of real-life settings. Competitive equilibrium (CE) is a central solution concept in economics to study markets, and due to its remarkable fairness and efficiency properties (e.g., envy-freeness, proportionality, core stability, Pareto optimality), it is also one of the most preferred mechanisms for fair division even though there is no money involved.

The vast majority of work in fair division focuses on the case of disposable goods, which all agents like or can throw away at no cost. In this paper, we consider the case of mixed manna under linear utilities where some items are positive goods liked by all agents, some are bads (chores) that no one likes, and remaining some agents like and others dislike. The recent work of Bogomolnaia et al. [13] initiated the study of CE in mixed manna. They establish that a CE always exists and maintains all the nice properties found in the case of all goods. However, computing a CE of mixed manna is genuinely harder than in the case of all goods due to the non-convex and disconnected nature of the CE set. Our main result is a polynomial-time algorithm for computing a CE of mixed manna when the number of agents or items is constant.

KEYWORDS

Fair division; Competitive equilibrium; Mixed manna

ACM Reference Format:
Jugal Garg and Peter McGlaughlin. 2020. Computing Competitive Equilibria with Mixed Manna. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Fair division studies the problem of allocating a set of items among a set of agents in a *fair* and *efficient* way. This age-old problem arises naturally in a wide range of real-life settings such as division of: family inheritance [36], partnership dissolutions, divorce settlements [15], spectrum allocation [26], seats in courses [9, 39], and computing resources in peer-to-peer platforms [29]. The formal study of fair division dates back to the cake cutting problem introduced in the seminal work of Steinhaus [40]. Since then it has been an active area of research in many disciplines.

Competitive equilibrium (CE) is one of the fundamental solution concepts in Economics to study markets, where prices and allocation are such that demand of items meets their supply when each agent gets her most preferred and affordable bundle. A competitive allocation not only achieves the standard notion of fairness called *envy-freeness*, where every agent weakly prefers their allocation over any other agents' allocation, but it is also Pareto optimal, a standard notion of economic efficiency. Due to these remarkable fairness and efficiency guarantees, a CE is one of the preferred solutions for fair division problems even though there may be no money involved in the latter case. The most prominent example is *competitive equilibrium with equal incomes* (CEEI) [42], which creates a market by giving one virtual dollar to every agent.

The vast majority of work in both Economics and Computer Science focuses on the case of *disposable* goods, i.e., items that agents enjoy, or at least can throw away at no cost. However, many situations contain *mixed manna* where some items are positive goods (e.g., cake), while others are undesirable bads (e.g., house chores and job shifts). Potentially, agents might disagree on whether a specific item is a good or bad. Examples include: dividing tasks among various team members, deciding teaching assignments between faculty, managing pollution among firms, or splitting assets and liabilities when dissolving a partnership.

Clearly, bads are *nondisposable* and must be allocated. At first glance, it seems that the tools and techniques developed for the case of all goods might apply, but the mixed manna case turns out to be significantly more complex. The recent pioneering works of Bogomolnaia et al. [13, 14] initiated the study of CE mixed manna. The authors show the existence of CE and that it retains all the desirable fairness and efficiency guarantees found in the case of all goods. Thus, [13] argues that a CE remains the best mechanism for fair division of a mixed manna.

However, CE of mixed manna possess some peculiar properties. Namely, [13] establishes that generally multiple CE exist, and the set of equilibria is non-convex and disconnected. In sharp contrast, in the case of all goods, the unique equilibrium (in utilities) is captured by a convex program. Designing fast algorithms for mixed manna, is an important open question – the abstract of Bogomolnaia et al. [13] mentions,[1]

> *. . . the implementation of competitive fairness under linear preferences in interactive platforms like SPLIDDIT will be more difficult when the manna contains bads that overwhelm the goods.*

1.1 Our Contribution

We offer an algorithm to compute all CE of mixed manna under linear utilities that runs in polynomial time when either the number of agents or the number of items is constant. We note that most theoretical work in fair division studies linear utility functions, and in practice popular online platforms like SPLIDDIT employ linear

*Research supported by the NSF CAREER Award 1942321.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

[1] Spliddit [1] is a user friendly online platform for computing fair allocation in a variety of problems, which have drawn tens of thousands of visitors in the last five years [31]. Spliddit uses linear utilities.

utilities as it provides a simple, intuitive way for users to express their preferences [13]. Further, most applications, at least in online settings, involve only a few agents. For this reason, our requirement that either the number of agents or the number of items remains constant for polynomial time computation seems reasonable in practice. To the best of our knowledge, our work provides the first polynomial time algorithm to compute a CE under any set of assumptions.

Interestingly, every fair division instance of mixed manna falls into one of the three distinct types: positive, null, and negative (see Section 2.1). In positive problems all agents benefit by receiving non-negative utility, e.g., dividing assets and liabilities when the value of assets outweighs the cost of the liabilities, while in negative instances the value of bad items overwhelms the goods and all agents share the burden of completing the undesirable tasks, e.g., splitting household chores. Null instance are knife edge cases where each agent receives zero utility. We offer a simple linear program to determine instance type. We note that positive and null instances are both polynomial time solvable, while the complexity of negative instances remains an intriguing open problem (see Section 3).

As both positive and null instances are polynomial time solvable, we focus mostly on negative instances. Our approach uses a novel cell decomposition technique, as in [22]. The main idea here is that n hyperplanes separate $\mathbb{R}^d$ into $O(n^d)$ non-empty regions or 'cells'. We choose a set of hyperplanes that ensures each cell corresponds to a unique set of optimal items for each agent, i.e., we know which subset of items an agent might purchase within any given cell. We determine if this configuration of optimal items for each agent admits an equilibrium by checking certain conditions and solving a max flow problem on specially designed network.

1.2 Related Work

The fair division literature is too vast to survey here, we refer the reader to the excellent books [15, 34, 37] and restrict our attention to only the most relevant work.

Competitive allocation of good manna is very well-understood. The celebrated Eisenberg-Gale convex program captures equilibrium when utility functions are homothetic, concave and monotone, which includes linear [24, 25]. The program maximizes the Nash welfare on all feasible allocations, and implies existence, uniqueness (in utilities), and polynomial time computation of a CE; there are faster algorithms for some special cases [23, 35, 43, 44].

Most of the work in fair division is focused on allocating a 'good' manna with a few exceptions of 'bad' manna [8, 15, 37, 41]. Recent pioneering papers of Bogomolnaia et al. [13, 14] are the first to study the case of mixed manna. To the best of our knowledge, there is no work exploring the computation of competitive allocation of a mixed manna even under linear utilities. A recent work [16] provides a polynomial time algorithm for computing a competitive allocation for bad manna under linear utilities when the number of agents (or bads) is constant. Our work generalizes the result of [16] for mixed manna.

The fair allocation of *indivisible* items is also an intensely studied problem for the case when all items are goods with a few recent exceptions [4–7, 32, 38]. Since the standard notions of fairness such as envy-freeness are not applicable, alternate notions have been defined for this case; see [10, 17, 18, 30, 33] for a subset of notable work and references therein. The Nash welfare continues to serve as a major focal point in this case as well, for which approximation algorithms have been obtained under several classes of utility functions including linear [2, 3, 12, 19–21, 27].

2 PRELIMINARIES

Let M be a set of m items we wish to divide among the set N containing n agents. Wlog, we assume there is a unit amount of each item. An allocation $x = (x_1, \ldots, x_n)$ assigns a bundle of items $x_i = (x_{i1}, \ldots, x_{im})$ to each agent $i \in N$, where $x_{ij} \in [0, 1]$ is the amount of item j given to agent i. A feasible allocation fully assigns each item between the agents, i.e., $\sum_i x_{ij} = 1, \forall j \in M$. Let $u_i : \mathbb{R}^m \to \mathbb{R}$ denote agent i's utility function. We assume all agents have linear utility functions. That is, $u_{ij} \in \mathbb{R}$ describes the utility agent i receives from item j, and i's utility for bundle $x_i \in \mathbb{R}^m$, is $u_i(x_i) = \sum_j u_{ij} x_{ij}$. Note that, unlike the traditional setting of all goods, some items $j \in M$ may be positive goods liked by all agents $u_{ij} > 0, \ \forall i \in N$, while other items $j \in M$ are universally disliked bads (chores) $u_{ij} \le 0, \ \forall i \in N$. Further, two agents i and i' might disagree on whether item j is a good or bad, i.e., $u_{ij} > 0$ and $u_{i'j} \le 0$, or vice versa.

The tuple $\mathcal{I} = \langle N, M, U \rangle$ defines a fair division instance, where $U = \{u_1, \ldots, u_n\}$ gives the agents' utility functions. We create a competitive division instance $\mathcal{I}' = \langle N, M, U, e \rangle$ by introducing virtual prices $p = (p_1, \ldots, p_m)$ for the items, and budgets $e = (e_1, \ldots, e_n)$ for the agents. Recall that the agents in the fair division instance $\mathcal{I}$ have no money. However, we require budgets and prices in terms of virtual currency to define both the competitive equilibrium solution concept, as well as our algorithm. We note that both item prices and agents' budgets may be negative, as discussed shortly. Despite this fact, we say agent i 'purchases' or 'spends' on item j if $x_{ij} > 0$, i.e., i receives some fraction of item j, and we say an agent i 'spends' her budget on bundle x_i if $\sum_j x_{ij} p_j = e_i$.

We note that, in all instances, all agents' budgets have the same sign, i.e., either $e_i \ge 0, \forall i \in N$, or $e_i < 0, i \in N$, which depends on the type of problem instance (see Section 2.1). Settings where agents possess different budgets represent situations where agents have different entitlements to the manna, e.g., splitting assets and liabilities when dissolving a partnership where one partner is more senior than another. We refer to the special case where all agents have the same budget as a Competitive Equilibrium of Equal Incomes (CEEI).

Definition 2.1. Given virtual budgets e, the pair (x, p) define a competitive equilibrium of a mixed manna if

- All agents spend their budgets: $\sum_j x_{ij} p_j = e_i, \forall i \in N$.
- All items are fully allocated: $\sum_i x_{ij} = 1, \forall j \in M$.
- Agents purchase optimal bundles of items at prices p: the bundle x_i solves

$$\begin{aligned} \max_{x_i \in \mathbb{R}^m} \quad & \sum_j u_{ij} x_{ij} \\ s.t. \quad & \sum_j x_{ij} p_j \le e_i \\ & x_{ij} \ge 0, \forall i, j. \end{aligned} \tag{1}$$

2.1 Results of Bogomolnaia et al. [13]

Bogomolnaia et al. [13] prove the existence of competitive equilibria under fairly general assumptions on agents utility functions. We provide a brief summary of their results below.

Definition 2.2. Given a fair division instance $\langle N, M, U \rangle$, define the following two types of agents:

- Agent i is attracted to the manna if $u_{ij} > 0$ for some item $j \in M$. Let $N^+ = \{i \in N : \exists j \in M, u_{ij} > 0\}$ be the set of attracted agents.
- Agent i is repulsed by the manna if $u_{ij} \leq 0, \forall j \in M$. Let $N^- = \{i \in N : u_{ij} \leq 0, \forall j \in M\}$ be the set of repulsed agents.

In words, attracted agents view some item $j \in M$ as a good, while repulsed agents view all items as undesirable chores. For any repulsed agent $i \in N^-$, any allocation where $x_i = \mathbf{0}$ maximizes her utility. The above definitions allow us to define the type of competitive division instance. Let $\mathcal{X}$ denote the set of feasible allocations, and let $\mathcal{U}$ be set of agent utilities over all feasible allocations, i.e., if $u \in \mathcal{U}$, then $u = (u_1(x_1), \ldots, u_n(x_n))$ for some $x \in \mathcal{X}$. Define the cone $\Gamma_+ = \mathbb{R}_+^{N^+} \times \{0\}^{N^-}$. Note that in Γ_+ attracted agents benefit without harming any repulsed agents. Also, let $\Gamma_{++} = \mathbb{R}_{++}^{N^+} \times \{0\}^{N^-}$ be the relative interior of Γ_+.

Definition 2.3. (Instance Type) Any given fair division instance $\langle N, M, U \rangle$ falls into one of the following three types of competitive division instance:

- If $\mathcal{U} \cap \Gamma_{++} \neq \emptyset$, then instance is positive.
- If $\mathcal{U} \cap \Gamma_+ = \{\mathbf{0}\}$, then instance is null.
- If $\mathcal{U} \cap \Gamma_+ = \emptyset$, then instance is negative.

In words, in a positive instance we can ensure all attracted agents receive strictly positive utility without harming any repulsed agents (who dislike all items). In a negative instance, no feasible allocation gives *all* attracted agents non-negative utility. In null instances, the only feasible allocations which give all agents non-negative utility satisfy $u_i(x_i) = 0, \forall i \in N$. The type of instance determines the form of the competitive equilibrium. Let $\mathcal{X}^*$ denote the set of Pareto optimal allocations, and define $\mathbb{R}_{--}^N$ as the cone where all agents receive strictly negative utilities.

Theorem 2.4 ([13]). *If agents' utility functions are linear, then a competitive equilibrium exists for all instance types. Specifically*

- *In a positive instance, there exists a unique (in utilities) CE (x^*, p^*) by setting $e_i = 1, \forall i \in N^+$, and $e_i = 0, \forall i \in N^-$. The allocation x^* maximizes the Nash social welfare over all agents $i \in N^+$*

$$\max_{x \in \mathcal{X}} \prod_{i \in N^+} u_i(x_i).$$

- *In a null instance, the exists a unique (in utilities) CE (x^*, p^*) by setting $e_i = 0$, $i \in N$, all prices $p_j = 0$. In the allocation x^* all agents receive bundle such that $u_i(x_i^*) = 0$.*
- *In a negative instance, there exists at least one CE (x^*, p^*) by setting $e_i = -1, \forall i \in N$. Further, a CE allocation x^* is a critical point of the function*

$$\prod_{i \in N} |u_i(x_i)|, \ \text{s.t.} \ x \in \mathcal{U} \cap \mathbb{R}_{--}^N. \tag{2}$$

Recalling the definitions of the various problem types offers an intuitive interpretation of Theorem 2.4. In positive instances, we can ensure all attracted agents receive strictly positive utility without harming any repulsed agents. Therefore, a CE allows attracted agents to compete to over the items, while repulsed agents take a bundle they value at zero (which maximizes their utility). In a negative instance, no feasible allocation assigns all attracted agents non-negative utility. A CE asks that all agents share the burden of completing the undesirable tasks in an efficient way. In null instances, the best we can do without harming any agent is give each agent zero utility, which is exactly the CE solution.

2.2 Fairness Notions

In this section, we present a number of standard fairness and efficiency notions applicable to divisible items.

Definition 2.5. **Envy-freeness:** An allocation x is envy-free (EF) if every agent i weakly prefers her bundle x_i

$$\forall i \in N, \ u_i(x_i) \geq u_i(x_j), \forall j \in N. \tag{3}$$

EF defines the gold standard of fairness metrics. In the case of additive valuations, EF also implies another notion of fairness.

Definition 2.6. **Proportionality:** An allocation x is proportional (PROP) if every agent $i \in N$ receives a $1/n$ share of the items, where $n = |N|$ is the number of agents

$$\forall i \in N, \ u_i(x_i) \geq \frac{1}{n} u_i(M). \tag{4}$$

One can verify EF implies PROP by summing (3) over all $i \in N$, and using the definition of linear utilities, i.e., $u_i(M) = \sum_j u_{ij}$.

Finally, we define the standard notion of economic efficiency, Pareto Optimality (PO). Informally, allocation x is PO if no other allocation x_i' helps one agent $u_i(x_i') > u_i(x_i)$, without harming another agent k, $u_k(x_k') < u_k(x_k)$.

Definition 2.7. **Pareto Optimal** The allocation x Pareto dominates the allocation x' if $u_i(x_i) > u_i(x_i')$ for some agent $i \in N$, while $u_k(x_k) \geq u_k(x_k'), \forall k$. The allocation x is Pareto optimal if no allocation x' Pareto dominates it.

EF and PROP represent standard notions of fairness in the setting of divisible goods, while PO defines the standard notion of economic efficiency, i.e., non-wastefulness of resources. [13] shows that a competitive equilibrium of mixed manna provides all of these guarantees simultaneously.

Theorem 2.8. *Every competitive equilibrium allocation is EF, PROP, and PO.*

Theorem 2.8 presents a compelling case for a competitive equilibrium as the 'most fair' outcome in a fair division instance, since it also satisfies multiple other fairness guarantees and they are non-wasteful (PO). For these reasons we seek to find CE as the solution to fair division of mixed manna.

2.3 Identifying Goods and Bads

Observe that examining the sign of $u_{ij} \in \mathbb{R}$, for all agents $i \in N$ determines the price of each item. If for some item $j \in M$ there exists an agent $i \in N$ such that $u_{ij} > 0$, then j is a good with $p_j > 0$, since agent i has infinite demand for item j at any negative price. Similarly, if $\max_i u_{ij} \leq 0$, then item $j \in M$ is a bad with $p_j \leq 0$, since there is no demand at *any* positive price. Further, if $\max_i u_{ij} = 0$ for any bad j, then *all* competitive equilibria set $p_j = 0$, and we may allocate j to *any* agent $i \in N$ such that $u_{ij} = 0$. In view of the above discussion, we refer to items with $p_j > 0$ as goods, and items with $p_j < 0$ as bads. Obviously, this assumes a preprocessing step to remove items j with $\max_i u_{ij} = 0$, which have price $p_j = 0$. Let M^+ be the set of goods, and M^- be the set of bads.

The signs of prices for goods and bads leads to the following natural economic interpretation. Suppose agent i accepts responsibility to complete a fraction $x_{ij} > 0$ of a universally disliked bad (chore) $j \in M^-$. As the price of j is negative, she 'spends' a negative amount of virtual currency, thereby reducing her total spending. Equivalently, she receives a payment in virtual currency to perform part of task no one wants to complete in order to 'earn' more money to spend on goods she likes.

2.4 Finding Optimal Bundles

Given a vector of virtual prices $p \in \mathbb{R}^m$, define agent i's bang per buck for good j as

$$bpb_{ij} = \frac{u_{ij}}{p_j}.$$

Similarly, for any agent i and any bad $j \in M$ define the i's pain per buck for bad j as

$$ppb_{ij} = \frac{u_{ij}}{p_k} > 0,$$

assuming we have removed all items where $\max_i u_{ij} = 0$. Observe that bpb_{ij} (ppb_{ij}) give agent i's utility (disutility) per unit spending on item j. Intuitively, the optimal bundles for agent i at the given prices p satisfy: i purchases only maximum *bpb* goods which give the highest utility per unit spending, and i purchases only minimum *ppb* items which give the lowest disutility per unit spending. These facts are easily verified by applying KKT conditions to (1).

For a given set of prices p, define the maximum bang per buck mbb_i goods for agent i as $mbb_i = \{j \in M^+ : u_{ij}/p_j = \max_k u_{ik}/p_k\}$. Similarly, define the minimum pain per buck mpb_i bads for i as $mpb_i = \{j \in M^- : u_{ij}/p_j = \min_{k \in M^-} u_{ik}/p_k\}$. In view of the above discussion, agent i only purchases mpb_i goods (if any), and mpb_i bads (if any). We summarize the cases where i purchases goods and bads in the special case of a negative instance below.

Proposition 2.9. *Let (x, p) be an allocation and price pair in a negative competitive division instance $\mathcal{I}$ with linear utilities.*

- *If $mbb_i = mpb_i$, then agent i potentially purchases both goods and bads.*
- *If $mbb_i < mpb_i$, then agent i purchases only bads and no goods.*
- *If $mbb_i > mpb_i$, then (x, p) is not a competitive equilibrium.*

Proof. In a negative instance, agents have negative budgets $e_i = -1, \forall i \in N$. Therefore, all agents must purchase some bads, and each unit of spending on goods must be offset by an equal amount of spending on bads. If $mbb_i < mpb_i$, then all goods give lower utility per unit spending the disutility incurred per unit spending on any bad. Therefore, i purchases no goods since agents try to maximize their utility. If $mbb_i > mpb_i$, then i gains utility by spending equal amounts on mbb_i goods and mpb_i bads. Thus, i has infinite demand for both mbb_i goods and mpb_i bads. □

For our algorithm and analysis later, we only need to specifically identify which bads and goods an agent purchases in a negative instance. One can derive similar results for positive problems, where budgets $e_i = 1, \forall i \in N$, by swapping the roles of mbb_i and mpb_i in Proposition 2.9. We note that the conditions of Proposition 2.9 are necessary but not sufficient for an equilibrium.

2.5 Utility per Buck Graph

We only require the following construction for negative instances, as defined in Section 2.1. As such, assume all agents only purchase items according to Proposition 2.9. Given prices p, we define the following bipartite graph $G(p) = (V, E)$ that we refer to as the utility per buck graph (UPB). We drop the price argument when the meaning is clear. We create a vertex for each agent $i \in N$ on one side and a vertex each item $j \in M$ on the other side. Next, we create the following edges: $(i, j), \forall j \in mpb_i, \forall i \in N$, and (i, j) for $j \in mbb_i$, $\forall i \in N$ such that $mpb_i = mbb_i$. Observe that we never create edges (i, j) between goods $j \in mbb_i$ when $mbb_i < mpb_i$ as required by Proposition 2.9. Therefore, in a negative instance, edges of the UPB connect any agent i to the only bads and goods (if any) that she might purchase in order to satisfy the optimal bundles condition.

For given prices p, we refer to the connected components of UPB(p) as a component of the market. Notice that within a component of the market, say C_k, all the agents only purchase the items of C_k, by Proposition 2.9, and the items of C_k are only purchased by the agents of C_k. Therefore, in a CE the sum of prices in a component equals the sum of the agents budgets in the component.

3 ALGORITHM

In this section, we present our algorithm to compute a competitive equilibrium of a mixed manna when agents have linear utility functions. First, we preform a few preprocessing steps:

- Identify the set of bads M^- and the set of goods M^+.
- Identify the attracted and repulsed agents, N^+ and N^- respectively.
- If there exists an agent $i \in N$ such that $u_{ij} = 0$ for some bad $j \in M^-$, assign j to i and set $p_j = 0$.

Observe the our preprocessing step means that $p_j < 0, \forall j \in M^-$, and that $u_{ij} < 0, \forall j \in M^-, \forall i \in N$. By Definition 2.2, this means that in all positive and null instances, see Definition 2.3, all repulsed agents receive no allocation.

Recall, from Theorem 2.4, a CE depends on problem type: positive, null, or negative. We can determine the problem type by solving the following linear program (LP):

$$\begin{aligned} \max \quad & t \\ s.t. \quad & \sum_j u_{ij} x_{ij} \geq t, \ \forall i \in N^+ \\ & \sum_{i \in N^+} x_{ij} = 1, \ \forall j \in M \\ & x_{ij} \geq 0, \ \forall i \in N^+, j \in M \\ & x_{ij} = 0, \ \forall i \in N^-, j \in M. \end{aligned} \tag{5}$$

Note that the solution t gives a lower bound on any attracted agent's utilities by the first set of constraints. The second set of constraints simply requires that all items are fully allocated, and the fourth set of constraints ensures all agents $i \in N^-$ receive utility $u_i(x_i) = 0$ in both positive and null instance types. Note that this assumes that we preformed a preprocessing step to remove all items j such that $\max_i u_{ij} = 0$, as discussed in Section 2.3. It follows that repulsed agents must receive no items in positive or null instances, see Section 2.1.

Proposition 3.1. *Let (t^*, x^*) be a solution to (5). The sign of t^* in the solution to (5) determines the competitive division type.*

- *If $t^* > 0$, then the instance is positive.*
- *If $t^* = 0$, then the instance is null.*
- *if $t^* < 0$, then the problem is negative.*

Proof. Note that x^* is feasible due to the second set of constraints, and $u_i(x_i^*) = 0$, $\forall i \in N^-$, by the fourth set of constraints. Therefore, we only need to check whether some feasible allocation gives all agents in N^+ positive utility or not. We consider each case, based on sign of t^*, separately. First, suppose $t^* > 0$. Then $u(x^*) \in \Gamma_{++} = \mathbb{R}_{++}^{N^+} \times \{0\}^{N^-}$, i.e., the problem is positive.

Next, consider the case of $t^* = 0$. We aim to show $\mathcal{U} \cap \Gamma_+ = \{\mathbf{0}\}$. For the sake of contradiction, suppose $t^* = 0$ and $\exists x \in \mathcal{X}$ such that $u_i(x_i) \geq 0$, $\forall i \in N^+$ and $u_k(x_k) > 0$ for some $k \in N^+$ while $u_{k'}(x_{k'}) = 0$ for some $k' \in N^+$. We now go through a sequence of steps to improve the allocation x for at least some agent that ultimately gives all attracted agents positive utility, contradicting that $t^* = 0$ is the greatest lower bound on attracted agents utility over all feasible allocations.

Step 1: Recall, by the definition of a good $j \in M^+$, $\exists i \in N^+$ such that $u_{ij} > 0$. If good $j \in M^+$ is partially assigned to agent k, i.e., $x_{kj} > 0$, and $u_{kj} \leq 0$, then allocate k's fraction of j to agent i, i.e., $x_{ij} \leftarrow x_{ij} + x_{kj}$ and $x_{kj} \leftarrow 0$. Clearly, this improves i's utility, without deducing k's utility.

Step 2: Consider any agent $i \in N^+$ with no zero allocation, $x_i \neq \mathbf{0}$, but $u_i(x_i) = 0$. Since $u_{ij} < 0$, $\forall j \in M^-$, and Step 1 ensures i holds no $j \in M^+$ that she values at $u_{ij} = 0$, then i holds some fraction of a good $j \in M^+$ she likes and some fraction of a bad $j' \in M^-$ she dislikes. By assumption, $u_k(x_k) = c > 0$ for some $k \in N^+$. Consider giving some fraction of j' to agent k

$$x_{kj'} \leftarrow x_{kj'} + \min\left(\frac{c}{2|u_{kj'}|}, x_{ij'}\right), \text{ and } x_{ij'} \leftarrow x_{ij'} - \min\left(\frac{c}{2|u_{kj'}|}, x_{ij'}\right).$$

After this transfer $u_k(x_k) > c/2 > 0$, and $u_i(x_i) > 0$.

Step 3: After Steps 1 and 2, either $u_i(x_i) > 0$, or $x_i = \mathbf{0}$, $\forall i \in N^+$. If $u_i(x_i) > 0$, $\forall i \in N^+$, then this contradicts that t^* is the solution to (5). Otherwise, $\exists i \in N^+$ with $x_i = \mathbf{0}$. Since $i \in N^+$, $\exists j \in M^+$, such that $u_{ij} = c > 0$. Observe that j is fractionally assigned to some $k \in N^+$, by Step 1. Consider reallocating some of j to agent i

$$x_{kj} \leftarrow x_{kj} - \min\left(\frac{c}{2u_{kj}}, x_{kj}\right), \quad \text{and} \quad x_{ij} \leftarrow \min\left(\frac{c}{2u_{kj}}, x_{kj}\right).$$

After this transfer, both $u_i(x_i), u_k(x_k) > 0$. Repeating this over all agents with $x_i = \mathbf{0}$ ensures $u_i(x_i) > 0$, $\forall i \in N^+$, contradicting $t^* = 0$ solves (5).

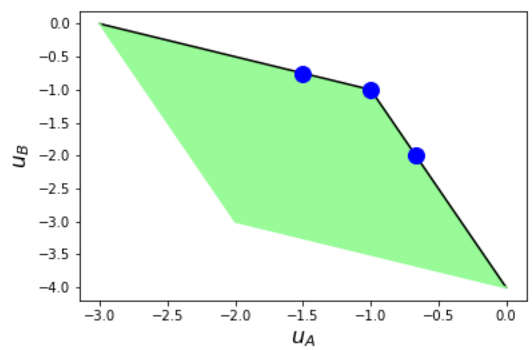

Figure 1: Agents utilities and all CE of Example 1.

Finally, the case of $t^* < 0$, the need to show no feasible allocations yield $u_i(x_i) \geq 0$, $\forall i \in N$. One can adapt the arguments for the $t^* = 0$ case for this purpose. □

Proposition 3.1 implies that both positive and null problems are polynomial time solvable. Indeed, if the problem is positive, i.e., the solution gives $t^* > 0$ in (5), then Theorem 2.4 shows that maximizing Nash welfare over all attracted agents N^+ gives a competitive equilibrium. Since log is a monotone increasing function, we may equivalently maximize the objective $\sum_{i \in N^+} \log(x_i)$ over the same constraints as (5). Therefore, any off the shelf convex optimization solver yields a competitive equilibrium from the solution x^* and prices p^* which correspond to dual variables of the first set of constraints. Similarly, if the problem is null, then x^* from the solution to (5) gives a feasible allocation such that all agents receive zero utility. Thus, setting all budgets e_i and prices p_j equal to zero yields the unique (in utilities) competitive equilibrium.

Theorem 3.2. *Let $\mathcal{I}$ be a fair division instance. If the problem is positive or null, then one can compute the unique (in utilities) CE in polynomial time. If the problem is negative, then one can compute all CE in polynomial time if either the number of agents or the number of items is constant.*

3.1 Handling Negative Problems

Negative problems present a more interesting challenge for computing an equilibrium. Theorem 2.4 establishes that equilibria are critical points of (2) on $\mathcal{U}^* \cap \mathbb{R}_{--}^N$, i.e., Pareto optimal allocations where all agents receives strictly negative utility. A simple example illustrates the difficulty.

Example 1: (Negative Problem) Consider a fair division instance with two agents A and B, and two items 1 and 2. The agents utility functions are: $u_A(x_A) = -x_{A1} - 2x_{A2}$, and $u_B(x_B) = -3x_{B1} - x_{B2}$. Observe both items are bads, so clearly no feasible allocation gives both agents non-negative utility. Therefore, the problem is a negative instance. Figure 1 plots the agents' utilities over all feasible allocations, shown as the green shaded region. The black curve shows the Pareto frontier $\mathcal{X}^*$, and the three equilibria are shown in as blue dots. Clearly, the set of equilibria are non-convex and disconnected.

This highlights the two major problems in computing a competitive equilibrium in negative instances. First, one needs to determine the Pareto frontier $\mathcal{X}^*$. [13, 14] offer a method to find $\mathcal{X}^*$ when there are either only two agents or only two items. However, no general method is known outside of these two very restrictive special cases. Second, unlike positive and null cases which have a unique (in utilities) equilibrium, a general negative case admits multiple equilibria. Moreover, different equilibria provide different utilities for the agents, introducing a new problem of equilibrium selection. One can argue that finding *all* equilibria might be necessary to ensure the fairest outcome. Consider Example 1, the three CE in Figure 1: right, middle, and left; give the agents utilities $(u_A(x_A), u_B(x_B))$ of: $(-3/2, -3/4)$, $(-1, -1)$, and $(-2/3, -2)$; respectively. Thus, the left (right) CE is most preferred by agent B (A). Each CE is envy-free, proportional, and PO, but one could argue that the middle CE is the most fair.

3.2 Algorithm for Negative Instances

We now present our algorithm to compute *all* competitive equilibria of a negative instance. Our approach relies on enumeration, and thus, has exponential runtime in the worst case. However, we show that if either the number of agents or the number of items remains constant, then our algorithm runs in polynomial time. We note that our algorithm relies on the characterization of optimal bundles in Section 2.4, rather than attempting to find critical points of (2) on $\mathcal{U}^* \cap \mathbb{R}^N_{--}$.

We use the 'cell decomposition' technique, as in [22]. The central concept is the fact that k hyperplanes in $\mathbb{R}^d$ form at most $O(k^d)$ non-empty regions, or cells. Suppose the number of items is fixed $m = |M|$. We consider the space $\mathbb{R}^m$ with coordinates corresponding to prices $p_1, \dots, p_m$. We create polynomially many hyperplanes to partition $\mathbb{R}^m$ into polynomially many cells, since m is fixed. Our choice of hyperplanes ensures that each cell corresponds to a unique configuration of mbb_i goods and mpb_i bads for each agent $i \in N$. Since optimal bundles, Section 2.4, require agents purchase only mbb_i and mpb_i items, we know what items any agent might purchase in a given cell. Then, it remains to check the other two conditions required for an equilibrium to hold: all agents spend their budgets, and total spending on any item equals its price. We show that solving a max flow problem on a certain network suffices. By checking all (polynomially many) cells, we find *all* competitive equilibria.

Before giving our cell decomposition to determine the set of mbb_i goods and mpb_i bads for each agent $i \in N$, we show how to use this information to compute an equilibrium (if one exists).

3.3 Finding Prices in a Cell

Our approach to computing an equilibrium in a given cell computes a max flow on a certain network. Before specifying the network, we start with a crucial lemma.

Lemma 3.3. *Given the set of mbb_i goods and mpb_i bads in a given cell, one can determine all prices.*

Proof. We use the utility per buck graph (UPB), defined in Section 2.5. Recall that, within a component C_k, all agents of C_k spend their budgets only on items of C_k, and all items of C_k are only purchased by the agents of C_k. Note that in a negative instance $e_i = -1$, $\forall i \in N$, and therefore each component contains at least one bad $j \in M^-$ with $p_j < 0$. Let $n_k = |N \cap C_k|$ be the number of agents in C_k. Any CE of a negative instance satisfies $-n_k = \sum_{i \in C_k} e_i = \sum_{j \in C_k} p_j$. We want to rewrite the price of all items in terms of some representative bad $j_0 \in C_k \cap M-$, then using $-n_k = \sum_{j \in C_k} p_j$, we can determine the price of all items in the component. Refer to Figure 2 in the following discussion.

In each component C_k, pick a representative bad $j_0 \in C_k \cap M^-$. Since C_k is a connected component, there exists a path connecting bad j_0 and all other items $j_\ell \in C_k$ in the UPB graph. This path alternates between items j_a and agents i_a: (j_0, i_0),(i_0, j_1), $\dots$, $(i_{\ell-1}, j_\ell)$. Each edge on this path is either mbb_a or mpb_a for the agent i_a, i.e., $u_{aj_a}/p_{j_a} = u_{aj_{a+1}}/p_{j_{a+1}}$, because we use the UPB graph.

For each component C_k, form a tree t with root $j_0 \in C_k \cap M^-$ by following the shortest alternating path between items j_0 and $j_\ell \in C_k$ in the UPB, including the agent i_a that connects items j_a and j_{a+1} along mpb_i or mbb_i edges. See Figure 2 for an illustration.

Observe that the leaves of the tree t correspond to items. For any leaf $\ell \in t$, we may write the price of j_ℓ in terms of the

Data: Negative competitive division instance $\langle N, M, U, e\rangle$
Result: All CE
Determine cells using Section 3.5 if constantly many items, or Section 3.6 if constantly many items;
$CE = \emptyset$;
for *each cell c* **do**
  Determine prices p in c as in Section 3.3;
  if *prices are consistent* **then**
    Form economic network in Section 3.4;
    Solve max flow f in the network;
    if $f = \sum_{j \in M^-} |p_j|$ **then**
      Set $x_{ij} = f_{ij}/p_j, \forall j \in M^+$;
      Set $x_{ij} = f_{ji}/|p_j|, \forall j \in M^-$;
      $CE = CE \cup (x, p)$;
    end
  end
end

Algorithm 1: Compute all CE of negative instance.

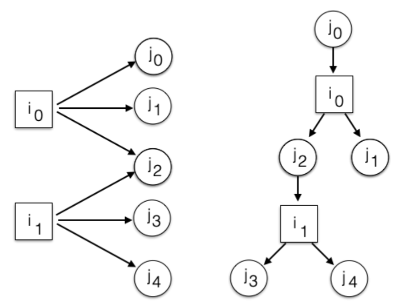

Figure 2: The UPB graph on the left, and its tree representation on the right.

representative good j_0 by following the tree up to the root, i.e., $p_\ell = p_{j_0} \prod_{k=1}^{\ell} u_{i_k j_k}/u_{i_{k-1} j_{k-1}} = p_{j_0} c_\ell$. Therefore, we have that $\sum_{j \in C_k} p_j = p_{j_0} \sum_{j \in C_k} c_j$. Finally, since in a CE we have $-n_k = \sum_{j \in C_k} p_j$, it follows that we can determine p_{j_0}, and therefore all $p_j \in C_k$.

Notice that the above procedure does not explicitly guarantee that prices are consistent with a cell configuration. That is, we need to check the definitions of mbb_i and mpb_i hold, e.g., $u_{ij}/p_j = u_{ik}/p_k$, $\forall j, k \in mbb_i$, $\forall i \in N$, and $u_{ij}/p_j > u_{ik}/p_k$, $j \in mbb_i$, $\forall k \notin mbb_i$. We check mpb_i for these prices similarly. Note that a cell also specifies whether $mbb_i < mpb_i$, or $mbb_i = mpb_i$ (the final case $mbb_i > mbp_i$ does not admit an equilibrium by Proposition 2.9). We check the appropriate condition similar to the above. We discard all cells with invalid prices. Obviously, this step runs in strongly polynomial time. □

Observe that this approach works with minimal changes when agents have different budgets, simply use $\sum_{i \in C_k} e_i$ instead of -n_k.

3.4 Max Flow to Check for CE

With valid prices in hand, we check if there exists an allocation where all agents spend their budgets and all items are fully sold by solving a max flow problem on a specially designed network. We create one vertex for each agent $i \in N$, and one vertex for each item $j \in M$. We refer to a vertex by the agent or item it represents. The vertices are arranged left to right as: source s, then all bads, then all agents, then all goods (if any), and finally the sink t, refer to Figure 3. In the max flow, we use spending $q_{ij} = x_{ij} p_j$, $\forall i, j$, instead of working the allocation x. Note that 'spending' on bads is negative $q_{ij} \leq 0$, $\forall j \in M^-$, $\forall i \in M^+$. Let (a, b) denote a directed edge from a to b and create the following edges:

- (s, j) with capacity $-p_j > 0$, $\forall j \in M^-$. The flow on this edge f_{sj} will represent the total spending on bad j by the agents, i.e., $f_{sj} = \sum_i |q_{ij}| = |p_j| \sum_i x_{ij}$.
- (j, i) with capacity ∞, $\forall j \in mpb_i$, $\forall i \in N$. The flow f_{ji} will represent the amount i spends on bad j, i.e., $f_{ji} = |q_{ij}| = x_{ij}|p_j|$.
- (i, j) with capacity ∞, $\forall j \in mbb_i$, $\forall i \in N$ such that $mbb_i = mpb_i$. The flow f_{ij} will represent the amount i spends on good j, i.e., $f_{ij} = q_{ij} = x_{ij} p_j$.
- (j, t) with capacity p_j, $\forall j \in M^+$. The flow f_{jt} will represent the total spending on good j, i.e., $f_{ij} = \sum_i q_{ij}$.
- (i, t) with capacity 1, $\forall i \in N$. The flow f_{it} represents the total spending of agent i, i.e., $f_{it} = |\sum_j q_{ij}|$.

We refer to the above construction as the economic network for the cell. See Figure 3 for an illustration in an example with only one good with $p_j = 2$ and one bad with $p_j = -4$. The edge's capacity is shown above the edge, and the amount of flow on the edge is shown in below.

Lemma 3.4. *A CE exists in a cell if and only if the max flow on the economic network equals* $sum_{j \in M^-} |p_j|$. *Further, if a CE exists in this cell, then the max flow f gives the CE allocation.*

Proof. Recall from Section 2.4 that the utility per buck graph (UPB) consists of ℓ connected components $\{C_k\}_{k=1}^{\ell}$. Within each component C_k, the agents of C_k only purchase the items of C_k, and

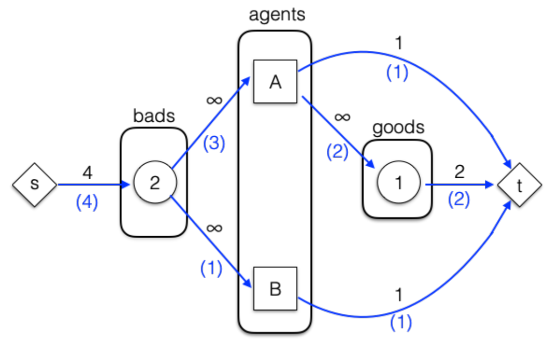

Figure 3: The network and max flow for Example 1.

the items of C_k are only purchased by the agents of C_k. Further, purchases are only made on mpb_i and mbb_i edges (assuming $mbb_i = mpb_i$ by Proposition 2.9) for each agent $i \in N$. Since we only add edges to the economic network under the same conditions, then the economic network consists of the same connected components as the UPB. Further, it suffices to check CE conditions hold within each component separately, i.e., agents in C_k spend their budgets and the items of C_k are fully sold. To avoid introducing additional notation, we assume there is only component, although the argument easily generalizes to the case of multiple components.

Recall in a negative instance, all agents budgets $e_i = -1$. Assuming the economic network is connected, then in a CE the sum of the agents budgets equals the sum of the item prices, i.e., $-n = \sum_i e_i = \sum_j p_j$, or $\sum_{j \in M^-} |p_j| = n + \sum_{j \in M^+}$. Observe from Figure 3 that max flow f is bounded above by $\sum_{j \in M^-} |p_j|$ where it saturates all edges leading out of the source (s, j), $\forall j \in M^-$. Similarly, f is bounded above by $\sum_{i \in N} |e_i| + \sum_{j \in M^+} p_j = n + \sum_{j \in M^+} p_j$ where it saturates all edges leading into the sink (i, t), $\forall i \in N$, and (j, t), $\forall j \in M^+$, $\forall i \in N$ such that $mbb_i = mpb_i$. Notice that our prices satisfy $\sum_{j \in M^-} |p_j| = n + \sum_{j \in M^+} p_j$. Further, the only edges out of $j \in M^-$ lead to agents $i \in N$ such that $j \in mpb_i$. Similarly, the only edges into $j \in M^+$ come from $i \in N$ such that $i \in mbb_i$ and $mbb_i = mpb_i$. By interpreting the flow f_{ji} from bad j to agent i as i's spending on j, i.e., $-f_{ji} = q_{ij} = x_{ij} p_j$, and f_{ij} as agent i's spending on good j, i.e., $f_{ij} = q_{ij} = x_{ij} p_j$, we obtain the following. By flow conservation, if the max flow $f = \sum_{j \in M^-} |p_j|$, then all items are fully purchased on mpb_i and mbb_i edges. Also, total flow into agent i comes from bads $j \in mpb_i$, and the total flow out goes to the sink t and goods $j \in mbb_i$. Therefore, if the max flow equals $\sum_{j \in M^-} |p_j| = n + \sum_{j \in M^+} p_j$, then $f_{it} = 1$, $\forall i \in N$. Then, flow conservation requires that for each agent i we have: $\sum_{j \in M^-} |q_{ij}| = \sum_{j \in M^-} f_{ij} = f_{it} + \sum_{j \in M^+} = 1 + \sum_{j \in M^+} q_{ij}$, or $e_i = -1 = \sum_j q_{ij} = \sum_j x_{ij} p_j$. □

Note that our approach easily generalizes to situations where agents have different budgets by simply changing the capacity on each edge (i, t) to be e_i. In view of Lemmas 3.3 and 3.4, we only need to determine all configurations of mbb_i goods and mpb_i bads in order to determine *all* equilibria of a negative instance.

3.5 Constant Number of Items

We seek a set of hyperplanes that uniquely determine the set of mbb_i and mpb_i items for each agent. Recall that our preprocessing step identifies the set of goods $j \in M^+$ with $p_j > 0$, and the set of bads $j \in M^-$ with $p_j < 0$. We begin by creating the hyperplanes $p_j = 0, \forall j \in M$. Each hyperplane divides $\mathbb{R}^m$ into half-spaces with signs >, =, and <, which correspond to $p_j > 0$, $p_j = 0$, and $p_j < 0$ respectively. Given all such hyperplanes, we refer to all non-empty regions as cells. Since $p_j > 0, \forall j \in M^+$, and $p_j < 0, \forall j \in M^-$, we say a cell is valid if the prices of goods (bads) are positive (negative). In the following, we only consider valid cells.

Next, we look for set of mbb_i goods and mpb_i bads within each valid cell. For this, we introduce hyperplanes $u_{ij}p_{j'} - u_{ij'}p_j = 0$, for all pairs of items $j, j' \in M$, and all $i \in N$. If both items are goods $j, j' \in M^+$, then in the > region of a hyperplane we have $u_{ij}/p_j > u_{ij'}/p_{j'}$, i.e., good j gives higher bang per buck than good j' for agent i. Similarly, in the = region both goods give the same bang per buck, while in the < region j' has better bang per buck than j. Proposition 3.5 summarizes similar results for cases when $j, j' \in M^-$, and when $j \in M^+$, and $j' \in M^-$.

Proposition 3.5. *Within any valid cell, the signs of hyperplanes $u_{ij}p_{j'} - u_{ij'}p_j = 0$ determine the following relations for each agent $i \in N$:*

- *If $j, j' \in M^+$ where $p_j, p_{j'} > 0$, then the sign >, =, < of the hyperplane means $bpb_{ij} > bpb_{ij'}$, $bpb_{ij} = bpb_{ij'}$, $bpb_{ij} < bpb_{ij'}$ respectively.*
- *If $j, j' \in M^-$ where $p_j, p_{j'}, u_{ij}, u_{ij'} < 0$, then sign >, =, < of the hyperplane means $ppb_{ij} > ppb_{ij'}$, $ppb_{ij} = ppb_{ij'}$, $ppb_{ij} < ppb_{ij'}$ respectively.*
- *If $j \in M^+$, and $j' \in M^-$, then sign >, =, < of the hyperplane means $bpb_{ij} < ppb_{ij'}$, $bpb_{ij} = ppb_{ij'}$, $bpb_{ij} > ppb_{ij'}$ respectively.*

It follows from Proposition 3.5 that the signs of the hyperplanes in each cell create a partial ordering on goods $j \in M^+$ in terms of bpb_{ij} and a partial ordering on the bads $j \in M^-$ in terms of ppb_i for each agent i. Thus we define mbb_i goods as the set with highest bpb_{ij}, and the mpb_i bads as the set with lowest ppb_i in each cell. Further, the hyperplanes which compare $j \in M^+$ and $j' \in M^-$ allow us to determine whether $mbb_i < mpb_i$, $mbb_i = mpb_i$, or $mbb_i > mpb_i$. Thus, by Proposition 2.9, we can determine which set of bads and goods (if any) an agent purchases in a cell. Notice that each cell gives a unique configuration of mbb_i goods, mpb_i bads, and the sign of $mbb_i \leq mpb_i$, or vice versa. Therefore, using the max flow approach of Section 3.4 we can compute the equilibrium in any given cell (if one exists). Observe that we created $\binom{m}{2}$ hyperplanes for each agent, and therefore $O(nm^2)$ in total. These hyperplanes divide $\mathbb{R}^m$ into at most $O((nm^2)^m)$ non-empty cells which is polynomially since the number of items is constant.

3.6 Constant Number of Agents

We use the same basic reasoning as before, this time exploiting the constant number of agents $n = |N|$. Recall that we need to create a set of hyperplanes that uniquely identify the set of mbb_i goods and mpb_i bads for each agent. For a given set of prices p, the mbb_i condition states that i purchases good $j \in M^+$ if and only if $u_{ij}/p_j = \alpha_i = \max_k u_{ik}/p_k$, and the mpb_i condition states that i purchases bad $j \in M^-$ if and only if $u_{ij}/p_j = \alpha_i = \min_k u_{ik}/p_k$. For each agent $i \in N$, we create a variable λ_i to serve as the reciprocal of i's mbb_i or mpb_i, i.e., $1/\lambda_i = \alpha_i$. Note that $\alpha_i > 0$, and so $\lambda_i > 0$.

Observe that $u_{ij}/p_j = 1/\lambda_i$ for any good $j \in mbb_i$, and $u_{ij}/p_j < 1/\lambda_i$ otherwise. Equivalently, $u_{ij}\lambda_i = p_j, \forall j \in mbb_i$, and $u_{ij}\lambda_i < p_j$ otherwise. Similarly, we have $u_{ij}\lambda_i = p_j$ for all bads $j \in mpb_i$, and $u_{ij}\lambda_i < p_j$ otherwise since $u_{ij}, p_j < 0$.

We consider the space $\mathbb{R}^n$ with coordinates $\lambda_1, \ldots, \lambda_n$. First we add the hyperplanes $\lambda_i = 0$, so that we only need to consider situations where $\lambda_i > 0$. We say a cell is valid if $\lambda_i > 0, \forall i \in N$. Next, we create the hyperplanes $u_{ij}\lambda_i - u_{i'j}\lambda_{i'} = 0$ for each pair of agents $i, i' \in N$, $\forall j \in M$. Within a cell, this gives a partial ordering on the terms $u_{ij}\lambda_i$ for each good $j \in M^+$. For each good $j \in M^+$, let $bpb^*_j = \{i \in N : u_{ij}\lambda_i = \max_k u_{kj}\lambda_k\}$ be the equivalence class of agents with the highest $u_{ij}\lambda_i$ values. Observe that, if $j \in M^+$, then in the > region we have $u_{ij}\lambda_i > u_{i'j}\lambda_{i'}$, or since $u_{ij}\lambda_i \leq p_j$, $1/\lambda_{i'} > u_{i'j}/p_j$, i.e., i' does not purchase good j. Since all items are fully assigned in a feasible allocation, we must have $j \in mbb_i$, $\forall i \in bpb^*_j$, and $mbb_i > bpb_{ij}$ for all other $i \in N$.

Similarly, for each bad $j \in M^-$, the hyperplanes give a partial ordering on the terms $u_{ij}\lambda_i$. Let $ppb^*_j = \{i \in N : u_{ij}\lambda_i = \max_k u_{kj}\lambda_k\}$ (since $u_{ij} < 0$ and $\lambda_i > 0$) be the equivalence class of agents with the highest $u_{ij}\lambda_i$ values for each bad $j \in M^-$. In the > region of any hyperplane, $u_{ij}\lambda_i > u_{i'j}\lambda_{i'}$, or since $u_{ij}\lambda_i \leq p_j$ and $p_j < 0$, we see that $u_{i'j}/p_j > 1/\lambda_{i'}$, i.e., i' does not purchase bad j. Thus, $j \in mpb_i$, $\forall i \in ppb^*_j$, and $mpb_i < ppb^*_j$ for all other $i \in N$.

From the above discussion, the signs of the hyperplanes $u_{ij}\lambda_i - u_{i'j}\lambda_{i'}$ within a cell give a unique configuration of mpb_i bads, and mbb_i goods for each agent $i \in N$. Therefore, we can determine if the cell admits a CE by computing the max flow on the network of Section 3.4. Observe that we created $\binom{n}{2}$ hyperplanes for each item, and therefore $O(mn^2)$ in total. These hyperplanes divide $\mathbb{R}^n$ into at most $O((mn^2)^n)$ non-empty cells which is polynomially since the number of agents n is constant.

4 DISCUSSION

We presented an algorithm to compute all CE of mixed manna under linear utilities that runs in polynomial time so long as either the number of agents or the number of items is constant. To the best of our knowledge, this first polynomial time algorithm under any set of assumptions. Our work also gives a simple LP to determine problem type, a method to determine the prices in each market component, and a new, specially designed network whose max flow proves that a CE exists for a given configuration of mbb_i and mpb_i items for each agent.

We see two interesting avenues for future work. First, our approach might generalize to more general classes of utility functions. Specifically, separable piecewise linear concave (SPLC) utility functions seems a natural candidate, see [22, 28] and references therein for more details. SPLC utilities are 'sufficiently close' to linear so that most of the basic structure of the algorithm and analysis should carry over. Second, despite the fact that both positive and null instances admit polynomial time algorithms, the complexity of computing a CE in a negative problem remains a major unresolved issue.

REFERENCES

[1] [n. d.]. . www.spliddit.org.
[2] Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. 2017. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. In *8th Innovations in Theoretical Computer Science Conference (ITCS)*. 1–12.
[3] Nima Anari, Tung Mai, Shayan Oveis Gharan, and Vijay V. Vazirani. 2018. Nash Social Welfare for Indivisible Items under Separable, Piecewise-Linear Concave Utilities. In *Proc. 29th Symp. Discrete Algorithms (SODA)*. 2274–2290.
[4] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. 2019. Fair Allocation of Indivisible Goods and Chores. In *Proc. 28th Intl. Joint Conf. Artif. Intell. (IJCAI)*.
[5] Haris Aziz, Hau Chan, and Bo Li. 2019. Maxmin Share Fair Allocation of Indivisible Chores to Asymmetric Agents. In *Proc. 18th Conf. Auton. Agents and Multi-Agent Systems (AAMAS)*. 1787–1789.
[6] Haris Aziz, Hau Chan, and Bo Li. 2019. Weighted Maxmin Fair Share Allocation of Indivisible Chores. In *Proc. 28th Intl. Joint Conf. Artif. Intell. (IJCAI)*.
[7] Haris Aziz, Gerhard Rauchecker, Guido Schryen, and Toby Walsh. 2017. Algorithms for Max-Min Share Fair Allocation of Indivisible Chores. In *Proc. 31st Conf. Artif. Intell. (AAAI)*. 335–341.
[8] Yaron Azrieli and Eran Shmaya. 2014. Rental harmony with roommates. *J. Economic Theory* 153 (2014), 128–137.
[9] Eric B. Budish and Estelle Cantillon. 2010. The Multi-Unit Assignment Problem: Theory and Evidence from Course Allocation at Harvard. *American Economic Review* 102 (2010).
[10] Siddharth Barman and Sanath Kumar Krishnamurthy. 2017. Approximation algorithms for maximin fair division. In *Proc. 18th Conf. Economics and Computation (EC)*. 647–664.
[11] Siddharth Barman and Sanath Kumar Krishnamurthy. 2019. On the Proximity of Markets with Integral Equilibria. In *Proc. 33rd Conf. Artif. Intell. (AAAI)*.
[12] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Finding Fair and Efficient Allocations. In *Proc. 19th Conf. Economics and Computation (EC)*. 557–574.
[13] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaia. 2017. Competitive Division of a Mixed Manna. *Econometrica* 85, 6 (2017), 1847–1871.
[14] Anna Bogomolnaia, Hervé Moulin, Fedor Sandomirskiy, and Elena Yanovskaia. 2019. Dividing bads under additive utilities. *Social Choice and Welfare* 52, 3 (2019), 395–417.
[15] Steven J. Brams and Alan D. Taylor. 1996. *Fair division - from cake-cutting to dispute resolution.* Cambridge University Press.
[16] Simina Branzei and Fedor Sandomirskiy. 2019. Algorithms for Competitive Division of Chores. (2019). arXiv:1907.01766.
[17] Eric Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *J. Political Economy* 119, 6 (2011), 1061–1103.
[18] Ioannis Caragiannis, David Kurokawa, Herve Moulin, Ariel Procaccia, Nisarg Shah, and Junxing Wang. 2016. The Unreasonable Fairness of Maximum Nash Welfare. In *Proc. 17th Conf. Economics and Computation (EC)*. 305–322.
[19] Bhaskar Ray Chaudhury, Yun Kuen Cheung, Jugal Garg, Naveen Garg, Martin Hoefer, and Kurt Mehlhorn. 2018. On Fair Division for Indivisible Items. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*. 25:1–25:17.
[20] Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay Vazirani, and Sadra Yazdanbod. 2017. Convex Program Duality, Fisher Markets, and Nash Social Welfare. In *Proc. 18th Conf. Economics and Computation (EC)*.
[21] Richard Cole and Vasilis Gkatzelis. 2015. Approximating the Nash Social Welfare with Indivisible Items. In *Proc. 47th Symp. Theory of Computing (STOC)*. 371–380.
[22] Nikhil Devanur and Ravi Kannan. 2008. Market Equilibria in Polynomial Time for Fixed Number of Goods or Agents. In *Proc. 49th Symp. Foundations of Computer Science (FOCS)*. 45–53.
[23] Nikhil Devanur, Christos Papadimitriou, Amin Saberi, and Vijay Vazirani. 2008. Market equilibrium via a primal–dual algorithm for a convex program. *J. ACM* 55, 5 (2008).
[24] Edmund Eisenberg. 1961. Aggregation of Utility Functions. *Management Sci.* 7, 4 (1961), 337–350.
[25] Edmund Eisenberg and David Gale. 1959. Consensus of subjective probabilities: The Pari-Mutuel method. *Ann. Math. Stat.* 30, 1 (1959), 165–168.
[26] R. Etkin, A. Parekh, and D. Tse. 2005. Spectrum Sharing for Unlicensed Bands. In *In Proceedings of the first IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*.
[27] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. 2018. Approximating the Nash Social Welfare with Budget-Additive Valuations. In *Proc. 29th Symp. Discrete Algorithms (SODA)*.
[28] Jugal Garg, Ruta Mehta, Milind Sohoni, and Vijay V. Vazirani. 2015. A Complementary Pivot Algorithm for Market Equilibrium under Separable, Piecewise-Linear Concave Utilities. *SIAM J. Comput.* 44, 6 (2015), 1820–1847.
[29] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. 2011. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. 323–336.
[30] Mohammad Ghodsi, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2018. Fair allocation of indivisible goods: Improvement and generalization. In *Proc. 19th Conf. Economics and Computation (EC)*. Available on arXiv:1704.00222 since April 2017.
[31] Jonathan R. Goldman and Ariel D. Procaccia. 2014. Spliddit: unleashing fair division algorithms. *SIGecom Exchanges* 13, 2 (2014), 41–46.
[32] Xin Huang and Pinyan Lu. 2019. An algorithmic framework for approximating maximin share allocation of chores. (2019). arXiv:1907.04505.
[33] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proc. 5th Conf. Economics and Computation (EC)*. 125–131.
[34] Herve Moulin. 2003. *Fair Division and Collective Welfare.* MIT Press.
[35] James Orlin. 2010. Improved algorithms for computing Fisher's market clearing prices. In *Proc. 42nd Symp. Theory of Computing (STOC)*. 291–300.
[36] John Winsor Pratt and Richard Jay Zeckhauser. 1990. The Fair and Efficient Division of the Winsor Family Silver. *Management Science* 36, 11 (1990), 1293–1301.
[37] J. Robertson and W. Webb. 1998. *Cake-Cutting Algorithms: Be Fair If You Can.* AK Peters, MA.
[38] Fedor Sandomirskiy and Erel Segal-Halevi. 2019. Fair Division with Minimal Sharing. *CoRR* abs/1908.01669 (2019).
[39] Tayfun Sönmez and Utku Unver. 2010. Course Bidding at Business Schools. *International Economic Review* 51, 1 (2010), 99–123.
[40] Hugo Steinhaus. 1948. The problem of fair division. *Econometrica* 16 (1948), 101–104.
[41] F. E. Su. 1999. Rental Harmony: Sperner's Lemma in Fair Division. *The American Mathematical Monthly* 106, 10 (1999), 930–942.
[42] Hal Varian. 1974. Equity, Envy and Efficiency. *J. Econom. Theory* 29, 2 (1974), 217–244.
[43] László Végh. 2014. Concave Generalized Flows with Applications to Market Equilibria. *Math. Oper. Res.* 39, 2 (2014), 573–596.
[44] László A. Végh. 2017. A Strongly Polynomial Algorithm for Generalized Flow Maximization. *Math. Oper. Res.* 42, 1 (2017), 179–211.