Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Complementary Pivot Algorithm for Competitive Allocation of a Mixed Manna

Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, Ruta Mehta

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–27
ISSN 0364-765X (print), ISSN 1526-5471 (online)
https://pubsonline.informs.org/journal/moor

A Complementary Pivot Algorithm for Competitive Allocation of a Mixed Manna

Bhaskar Ray Chaudhury,[a] Jugal Garg,[a,*] Peter McGlaughlin,[a] Ruta Mehta[a]

[a] University of Illinois at Urbana-Champaign, Urbana, Illinois 61801
*Corresponding author

Contact: braycha@illinois.edu (BRC); jugal@illinois.edu, https://orcid.org/0000-0001-6439-7308 (JG); mcglghl2@illinois.edu (PM); rutameht@illinois.edu (RM)

Received: April 5, 2021
Revised: May 19, 2022
Accepted: July 31, 2022
Published Online in Articles in Advance: September 26, 2022

MSC2020 Subject Classification: Primary: 91B50; 90C33; 68Q25

https://doi.org/10.1287/moor.2022.1315

Abstract. We study the fair division problem of allocating a mixed manna under additively separable piecewise linear concave (SPLC) utilities. A mixed manna contains goods that everyone likes and bads (chores) that everyone dislikes as well as items that some like and others dislike. The seminal work of Bogomolnaia et al. argues why allocating a mixed manna is genuinely more complicated than a good or a bad manna and why competitive equilibrium is the best mechanism. It also provides the existence of equilibrium and establishes its distinctive properties (e.g., nonconvex and disconnected set of equilibria even under linear utilities) but leaves the problem of computing an equilibrium open. Our main results are a linear complementarity problem formulation that captures all competitive equilibria of a mixed manna under SPLC utilities (a strict generalization of linear) and a complementary pivot algorithm based on Lemke's scheme for finding one. Experimental results on randomly generated instances suggest that our algorithm is fast in practice. Given the PPAD-hardness of the problem, designing such an algorithm is the only non–brute force (nonenumerative) option known; for example, the classic Lemke–Howson algorithm for computing a Nash equilibrium in a two-player game is still one of the most widely used algorithms in practice. Our algorithm also yields several new structural properties as simple corollaries. We obtain a (constructive) proof of existence for a far more general setting, membership of the problem in PPAD, a rational-valued solution, and an odd number of solutions property. The last property also settles the conjecture of Bogomolnaia et al. in the affirmative. Furthermore, we show that, if the number of either agents or items is a constant, then the number of pivots in our algorithm is strongly polynomial when the mixed manna contains all bads.

Funding: Financial support from the Division of Computing and Communication Foundations, National Science Foundation (NSF) [Grants 1942321, 1750436] is gratefully acknowledged.

Keywords: fair division • mixed manna • competitive equilibrium • LCP formulation • complementary pivot algorithm

1. Introduction

Fair division is the problem of allocating a set of items among a set of agents in a *fair* and *efficient* way. This age-old problem, mentioned even in the Bible, arises naturally in a wide range of real-life settings, such as division of family inheritance (Pratt and Zeckhauser [43]), partnership dissolutions, divorce settlements (Brams and Taylor [6]), spectrum allocation (Etkin et al. [25]), airport traffic management (Vossen [56]), office space between coworkers, seats in courses (Budish and Cantillon [8], Sönmez and Unver [47]), computing resources in peer-to-peer platforms (Ghodsi et al. [32]), and sharing of earth observation satellites (Bataille et al. [2]). The formal study of this problem dates back to the seminal work of Steinhaus [49] in which he introduced the cake-cutting problem for more than two agents. Since then, it has been an active research area in many disciplines.

The vast majority of work focuses on the case of *disposable* goods, that is, items that agents enjoy or at least can throw away at no cost. However, many situations contain *mixed manna* in which some items are positive goods, whereas others are undesirable bads (chores). Potentially, agents might disagree on whether a specific item is a good or a bad. Examples include dividing tasks among various team members, deciding teaching assignments between faculty, 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 good manna (i.e., no bads) might apply, but the mixed manna case turns out to be significantly more complex. The seminal work of Bogomolnaia et al. [3] initiated the study of mixed manna, in which they argue why allocating a mixed manna is genuinely more complicated than a good or a bad manna and why an allocation based on competitive equilibrium with equal incomes (CEEI) is the best mechanism. For example, competitive allocation not only achieves the standard notions of fairness called *envy-freeness* and *proportionality*, but it is also (Pareto) efficient and core stable. They show the existence of equilibrium and investigate some of its distinctive properties. Namely, they establish that even the simplest case of linear utility functions generally admits multiple equilibria, and the set of equilibria is nonconvex and disconnected.[1] In sharp contrast, in the same setting with good manna, an equilibrium is captured by a convex program. Designing fast algorithms for mixed manna, even for linear utilities, is an important open question; the abstract of Bogomolnaia et al. [3, p. 1847] mentions,[2] "… 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."

Recently, Branzei and Sandomirskiy [7] and Garg and McGlaughlin [27] have made progress on this problem by designing polynomial-time algorithms for computing competitive allocation under linear utilities when the number of either agents or items is a constant. These algorithms are based on clever enumeration-based exhaustive search, which may not be fast in practice in the general case.

1.1. Our Contributions

In this paper, we design a simplex-like algorithm for computing a competitive allocation of a mixed manna when agents' utility functions have a fairly general form: separable piecewise linear concave (SPLC), a strict generalization of linear; see Section 3 for a formal definition. In economics, it is customary to assume that utility functions of goods are concave because they capture the important condition of decreasing marginal utilities. Likewise, this assumption is also natural for bads to capture increasing marginal disutility, for example, considering the chore of reducing pollution from a plant at which driving emissions toward zero likely comes at a rising cost. The SPLC functions are also important for the fair division problems to capture natural situations when there are limitations on the maximum amount of an item that can be assigned to an agent because of rationing and other restrictions.

Experimental results on randomly generated instances suggest that our algorithm is fast in practice, answering the question raised by Bogomolnaia et al. [3]. Our algorithm follows a systematic path rather than a brute force enumeration of every *configuration*; see Section 5. The equilibrium computation problem is known to be PPAD-hard (Chen and Teng [11], Chen et al. [13], Vazirani and Yannakakis [53]) even when all items are goods. As a result, a polynomial time algorithm is not possible unless PPAD = P. We note that SPLC utilities are extensively studied in the case of good manna; see, for example, Chen and Teng [11], Chen et al. [13], Garg et al. [31], and Vazirani and Yannakakis [53]. To the best of our knowledge, they have not been studied before for a bad (or mixed) manna. We also note that Bogomolnaia et al. [3, 4] mention reducing the bads under linear utilities into goods under SPLC utilities; however, this may not always work; see Appendix A.

Our approach is based on Lemke's [38] complementary pivoting on a polyhedron, which is similar in spirit to a simplex algorithm for linear programming (Dantzig [16]) and the classic Lemke–Howson algorithm for computing a Nash equilibrium of a two-player game (Lemke and Howson [39]). A common phenomenon in these algorithms is that they perform well in practice even though their worst case behavior is exponential; the latter is exhibited via intricately doctored up instances that are designed to make the algorithm perform poorly; for example, see Klee and Minty [36] and Savani and von Stengel [46] for simplex and Lemke–Howson, respectively. Given the PPAD-completeness of our problem, such a pivoting-based algorithm is the only non–brute force (non-enumerative) option known.

The most striking feature of this approach is that it not only gives a fast algorithm, but also provides several new structural results as simple corollaries. First, it yields the first (constructive) proof of the existence of a competitive allocation of a mixed manna under SPLC utilities. Second, it shows that a rational-valued equilibrium exists if all input parameters are rational. Third, together with the result of Todd [51], it gives a proof of membership of this problem in PPAD. Fourth, this shows that the number of equilibria is odd in a nondegenerate instance. We note that none of these results were known even for linear utilities. The last property also settles the conjecture of Bogomolnaia et al. [3] in the affirmative, which shows the odd property for two agents (or two items) under linear utilities and conjectures the same for any number of agents and items.

Furthermore, we show that, if either the number of agents or the number items is a constant, then the number of pivots in our algorithm is polynomial when the mixed manna contains only bads. All our results also extend to a more general setting of *exchange*; see Section 2 for a definition. To the best of our knowledge, the exchange setting was not studied before despite its natural applications, for example, exchange of tasks among agents in which a group of university students is teaching subjects or sports to each other or some landlords providing shelter to apartment seekers in their houses in exchange for help in household chores; see, for example, www.mitwohnen.org.

1.2. Techniques

Our approach requires two steps. First, we need to derive a linear complementarity program (LCP) formulation for the problem whose solutions capture competitive equilibria. Second, we must show that the algorithm always terminates at a competitive equilibrium; this is usually shown by proving no *secondary rays* (a special kind of unbounded edges) in the LCP polyhedron; see Section 2.3 for details.

This approach is extensively utilized for computing equilibria in markets (with only goods) and in games; see, for example, Eaves [22], Garg and Vazirani [28], Garg et al. [30, 31], Koller et al. [37], Hansen and Lund [34], and Sørensen [48]. Each of them first obtains an LCP formulation that *exactly* captures equilibria and then shows that there are no secondary rays. Despite significant efforts, no such LCP was found for competitive allocation of mixed manna. Our LCP formulation has "nonequilibrium" solutions, and furthermore, it has secondary rays.

We first note that both the preceding steps must work *simultaneously*. In fact, it is not difficult to come up with an LCP formulation for only bads by extending the LCP for only goods (Eaves [22], Garg et al. [31]). However, it does not yield an algorithm. Hence, we first come up with a different LCP for only bads. The case of mixed manna turns out to be even more challenging as simply merging the two LCPs does not work. This is due to the single utility maximization over all items for each agent, and it is a priori not clear how much an agent wants to spend on only goods (or bads). Using new ideas, we derive an LCP formulation that captures competitive allocation of a mixed manna, but it also captures some nonequilibrium solutions that we deal with in the second step.

The second step presents the most significant challenge. The major issue with Lemke's scheme is that, in general, it is not guaranteed to find a solution. This happens when the path followed by the algorithm leads to a secondary ray.

As mentioned before, the standard way to show convergence of a complementary pivot algorithm to a solution is by proving that there are no secondary rays in the LCP polyhedron. However, our LCP formulation has secondary rays. Therefore, we must show that the algorithm never reaches a secondary ray to guarantee its termination to a competitive equilibrium. In addition, we must show that the final output of the algorithm is an equilibrium rather than a nonequilibrium solution to the LCP. This makes the analysis of our algorithm more challenging than the previous works.

1.3. Further Related Work

The fair division literature is too vast to survey here, so we refer to the excellent books Brams and Taylor [6], Moulin [40], and Robertson and Webb [44] and restrict attention to previous work that appears most relevant.

Most of the work in fair division is focused on allocating a good manna with a few exceptions of bad manna (Azrieli and Shmaya [1], Brams and Taylor [6], Robertson and Webb [44], Su [50]). The seminal paper of Bogomolnaia et al. [3] is the first to study the case of mixed manna. Whereas linear is the most studied utility function to model agents' preferences (Bogomolnaia et al. [3]), SPLC is its natural extension to capture important generalizations. For these models, competitive allocation of a good manna is very well-understood.

The two most ideal economic models to study competitive allocations are Fisher and Exchange. Fisher is a well-studied special case of the exchange model. In the Fisher setting, the celebrated Eisenberg–Gale convex program captures equilibrium when utility functions are homothetic, concave, and monotone, which includes linear (Eisenberg [23], Eisenberg and Gale [24]). The program maximizes the product of the agents' utilities (i.e., the Nash welfare) on all feasible utility profiles and implies existence, convexity, uniqueness (of utility profile), and polynomial time computation; there are faster algorithms for some special cases (Devanur et al. [19], Orlin [42], Végh [54, 55]). For exchange, polynomial time algorithms are known for subclasses of homothetic functions, including linear (Duan and Mehlhorn [20], Duan et al. [21], Garg and Végh [29], Jain [35], Ye [58]). Although the SPLC case is known to be PPAD-complete even in the Fisher setting (Chen and Teng [11], Chen and Teng [12]), the complementary pivot algorithm (Garg et al. [31]) works well in practice and is the only non–brute force option known.

As in Bogomolnaia et al. [3], we assume that agents' disutility values for bads (chores) are finite in this paper; see Section 2.1. We note that the complexity of computing equilibrium in our model under linear utilities is not settled yet. On the other hand, Chaudhury et al. [9] consider a slight variant of the problem in which an agent's disutility value for a bad can be infinity. For this model in the exchange setting, they show that computing a competitive allocation of a bad manna under *linear* utilities is already PPAD-hard. This result, together with the nonconvex and disconnected set of solutions, suggests that our algorithm in this paper is likely to be the best one can hope for this problem even under linear utilities.

A preliminary version of our work appeared in Chaudhury et al. [10].

The rest of the paper is organized as follows. We introduce notation and preliminaries in Section 2. In Section 3, we derive an LCP formulation that captures all competitive equilibria of a mixed manna under SPLC utilities. Our algorithm and its analysis appear in Section 4. A precise description of all the results is presented in Section 4.2. In Section 5, we show a strongly polynomial bound of the algorithm for only bad manna when the number of agents (or items) is a constant. Section 6 summarizes our numerical experiments on randomly generated instances. Appendix A presents a counterexample showing that bads cannot be reduced into goods, Appendix B illustrates that Lemke's scheme fails if we try a naive adaption of the LCP of Eaves [22] and Garg et al. [31], which is specialized to good manna. Finally, Appendix C shows the convergence of the algorithm for the special case of bad manna.

2. Preliminaries

Let M be the set of m divisible items that needs to be divided among the set N of n agents. An item can be a good or a bad (chore) for an agent as discussed earlier. Each agent i has a utility function $u_i : \mathbb{R}_+^m \to \mathbb{R}$ over bundles of items. Let $\boldsymbol{x}_i = (x_{ij})_{j \in M}$ denote agent i's assigned bundle containing x_{ij} amount of item j. The standard notions of fairness and efficiency are envy-freeness and *Pareto optimality*, defined as follows:

• (Weighted) envy-freeness: An allocation $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is said to have no envy if each agent weakly prefers the agent's allocation over any other agent's allocation, that is, $u_i(\boldsymbol{x}_i) \geq u_i(\boldsymbol{x}_j),\ \forall i, j \in N$.

When agents have different weights (unequal rights/responsibilities), say η_i is the weight of agent i, then we say that an allocation X has no envy if $\frac{u_i(x_i)}{\eta_i} \geq \frac{u_i(x_j)}{\eta_j},\ \forall i, j \in N$.

• Pareto optimality: An allocation $X' = (\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n)$ Pareto dominates another allocation $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ if $u_i(\boldsymbol{x}'_i) \geq u_i(\boldsymbol{x}_i)$ for all $i \in N$ and $u_k(\boldsymbol{x}'_k) > u_k(\boldsymbol{x}_k)$ for some $k \in N$. An allocation X is Pareto optimal if no allocation X' dominates X.

2.1. Utility Functions

In this paper, we consider additively SPLC utility functions, which are strict generalizations of linear.

In the case of linear utilities, $u_i(\boldsymbol{x}_i) := \sum_j U_{ij} x_{ij}$, where U_{ij} is the utility of agent i for a unit amount of item j. Clearly, $U_{ij} \geq 0$ if item j is a good for I, and $U_{ij} < 0$ if it is a bad. For bads, we also use $D_{ij} := |U_{ij}| > 0$ to denote the *disutility* of agent i for a unit amount of bad j.

In the case of SPLC utilities, $u_i(\boldsymbol{x}_i) = \sum_{j \in M} u_{ij}(x_{ij})$, where, for each agent i and each item j, the function $u_{ij} : \mathbb{R}_+ \to \mathbb{R}$ is monotone piecewise linear and concave. The function is either nonnegative and increasing representing a *good* or it is nonpositive and decreasing representing a *bad (chore)*. We call each linear piece of u_{ij} a segment. Let $|u_{ij}|$ be the number of segments of u_{ij}, and let the triple (i, j, k) denote the kth segment. The slope of a segment gives the utility received per each additional unit of the item. Let (i, j, k) be a segment with domain $[a, b] \subseteq \mathbb{R}_+$ and slope c. Define $U_{ijk} := c$, and $L_{ijk} := b - a$. Note that the length of the last segment is infinite. However, because there is unit amount of each item, we can assume without loss of generality that the length of the last segment is one plus some small constant. Note that linear is a special case of SPLC in which each u_{ij} has exactly one segment with infinite length. We assume that all U_{ijk}s are finite.

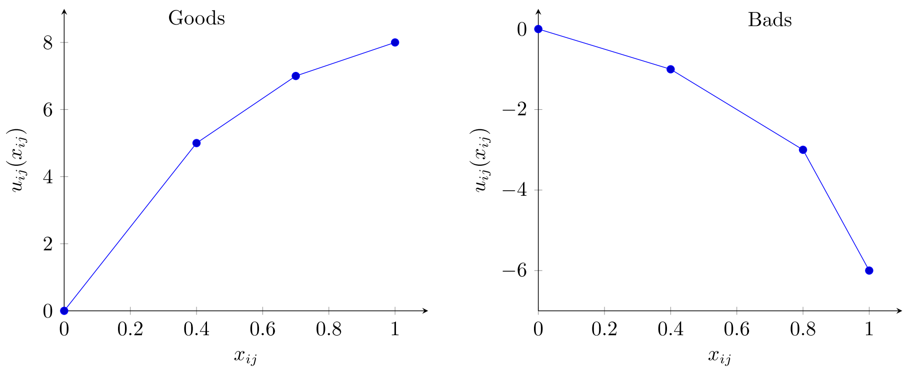
Our assumptions on the function u_{ij} imply the following. If agent i receives positive utility from item j, then $U_{ijk} > U_{ijk'} \geq 0$ for all $k < k'$, capturing the standard economic assumption of decreasing marginal returns on goods. Otherwise, $0 \geq U_{ijk} > U_{ijk'}$ for all $k < k'$, which models scenarios in which the disutility of completing a chore increases with the percentage required to be performed, for example, cutting emissions from a plant. In the latter case, we use the notation $D_{ijk} = |U_{ijk}|$ for agent i's disutility on the kth segment of u_{ij}. Figure 1 provides an illustration of SPLC utility functions.

2.2. Competitive Equilibrium

The two most ideal economic models to study competitive allocations are Fisher and exchange. These are two fundamental economic models, introduced by Fisher (Brainard and Scarf [5]) and Walras [57] in the late 19th century, respectively. An exchange model is like a barter system, in which each agent comes with an initial endowment of items and exchanges them with others to maximize the agent's utility function. Fisher is a special case of the exchange model in which each agent has a fixed proportion of each item. CEEI (Varian [52]) is further a special case of Fisher in which each agent has the same endowment.

Let $\boldsymbol{w}_i = (W_{ij})_{j \in M}$ denote agent i's initial endowment containing $W_{ij} \geq 0$ amount of item j. In Fisher, $W_{ij} = \eta_i$, $\forall i \in N, j \in M$, where η_i is the budget (entitlement/weight) of agent i. In CEEI, $\eta_i = 1/n,\ \forall i \in N$. Given prices of items, each agent demands a utility-maximizing (optimal) bundle by spending the agent's budget (earned by selling the initial endowment).

Figure 1. An example of SPLC utility functions for a good and a bad.



Definition 1 (Competitive Equilibrium). A competitive equilibrium is defined by prices $\boldsymbol{p} = (p_j)_{j\in M}$ and allocation $(\boldsymbol{x}_i)_{i\in N}$ satisfying the following two conditions:

1. Optimal bundle: Allocation $\boldsymbol{x}_i$ maximizes agent i's utility at $\boldsymbol{p}$, that is, $\boldsymbol{x}_i \in \{\arg\max u_i(\boldsymbol{y})$ s.t. $\sum_{j\in M} y_j p_j = \sum_{j\in M} W_{ij}p_j; y_j \geq 0,\ \forall j\}$. We note that it is without loss of generality to assume that agents exhaust their entire budgets at equilibrium for our sufficiency condition (defined subsequently) to ensure equilibrium existence in which each agent is nonsatiated for some good.

2. Demand meets supply (market clearing): Demand of each item equals its supply, that is, $\sum_{i\in N} x_{ij} = \sum_{i\in N} W_{ij}$, $\forall j \in M$.

Observe that equilibrium prices are scale invariant; that is, if $\boldsymbol{p}$ is an equilibrium price vector, then so is $\alpha\boldsymbol{p}$, $\forall\alpha > 0$. We can assume without loss of generality that each agent brings some fraction of some item and there is a unit amount of each item, that is, $\sum_{i\in N} W_{ij} = 1,\ \forall j \in M$. This is like redefining the unit of items by appropriately scaling utility values. Competitive allocations are well-known to be not only envy-free and Pareto optimal, but also core stable (i.e., no coalition of agents by standing alone can allocate better shares to each agent in the coalition). We note that competitive allocations may not imply Pareto optimality when agents are satiated. Because we assume that each agent is nonsatiated (defined subsequently) for some good to ensure equilibrium existence, it is always implied in our case.

2.2.1. Assumptions. Even in the special case of good manna, equilibria in an exchange setting need not exist (Devanur et al. [18]). We need to assume certain sufficiency conditions to allow an equilibrium to exist. We note that our conditions follow the previous works of Chen and Teng [11], Chen et al. [13], and Garg et al. [31] that consider only goods and is one of the weakest sufficiency conditions to guarantee that an equilibrium exists in the case of good manna. First, we include our basic assumptions.

Condition 1. *Each agent brings a positive amount of some good and positive amount of some bad.*

Definition 2 (Nonsatiation and Economy Graph). For any good $j \in M$, we say that agent i is *nonsatiated* for j if $U_{ijk} > 0$, where k is the last segment of good j. Define the economy graph as a directed graph G with vertices N with directed edges from i to i' if agent i is nonsatiated for some good j that agent i' brings. We call the instance *strongly connected* if the economy graph G is strongly connected.

Condition 2. *The economy graph of the input instance is strongly connected.*

Note that, Condition 2 is needed to ensure the existence of equilibrium even for the case of good manna (Chen et al. [13], Garg et al. [31]). We refer to Conditions 1 and 2 together as strong connectivity. We show that our algorithm in Section 4 converges to a competitive equilibrium under strong connectivity; hence, we get a constructive proof of the existence. Observe that this implies the existence of equilibrium in all instances of the Fisher (and, hence, CEEI) setting under linear utilities and nonsatiated SPLC utilities.

2.3. Linear Complementarity Problem and Lemke's Scheme

The LCP is a generalization of linear programming (LP) complementary slackness conditions: given an $n \times n$ matrix A and an n-dimensional vector $\boldsymbol{q}$, the problem is to find $\boldsymbol{y}$ such that

$$\forall i \in [n]: \ (A\boldsymbol{y})_i \leq q_i;\ \ y_i \geq 0;\ \ y_i(A\boldsymbol{y} - \boldsymbol{q})_i = 0, \tag{1}$$

where $[n] := \{1, 2, \ldots, n\}$. Clearly, the problem is only interesting when $q_j < 0$ for some $j \in [n]$; otherwise, $\boldsymbol{y} = \boldsymbol{0}$ offers a trivial solution. Let $\mathcal{P}$ denote the n-dimensional polyhedron defined by the first two constraints of (1). We assume that $\mathcal{P}$ is nondegenerate. That is, exactly $n - d$ constraints hold with equality on any d dimensional face of $\mathcal{P}$. Under this assumption, each solution to (1) corresponds to a vertex of $\mathcal{P}$ because exactly n equalities must be satisfied.

LCPs are general enough to capture (strongly) NP-hard problems (Cottle et al. [15]) and, therefore, may not have a solution. Lemke's scheme first *augments* the LCP by adding a scalar variable z to create easily accessible solutions and considers the formulation

$$\begin{array}{c}\forall i \in [n]: \quad (A\boldsymbol{y})_i - z \le q_i; \;\; y_i \ge 0; \;\; y_i((A\boldsymbol{y} - \boldsymbol{q})_i - z) = 0 \\ z \ge 0.\end{array} \tag{2}$$

Observe that a solution $(\boldsymbol{y}, z)$ with $z = 0$ of (2) gives a solution $\boldsymbol{y}$ of (1) and vice versa. Let $\mathcal{P}'$ be the polyhedron defined by the first two linear constraints for each $i \in [n]$ and $z \ge 0$ constraint. The dimension of $\mathcal{P}'$ is $n+1$. Assuming that $\mathcal{P}'$ is nondegenerate, solutions to (2) must still satisfy n constraints. Therefore, the set of solutions S is a subset of the one-skeleton of $\mathcal{P}'$; that is, solutions consist of edges (one-dimensional faces) and vertices (zero-dimensional faces) of $\mathcal{P}'$. Further, any solution to (1) must be a vertex of $\mathcal{P}'$ with $z = 0$.

Solutions S to the augmented LCP have some important structural properties. We say that label i is present at $(\boldsymbol{y}, z) \in \mathcal{P}'$ if $y_i = 0$ *or* $(A\boldsymbol{y})_i - z = q_i$. Every solution in S is *fully labeled* because label i is present for all $i \in [n]$. A solution $s \in S$ contains a *double label* i if $y_i = 0$ *and* $(A\boldsymbol{y})_i - z = q_i$ for $i \in [n]$. Further, there are two edges of S incident to s because there are only two ways to relax the double label still keeping all the other labels. Obviously, any solution s to (2) that satisfies $z = 0$ contains no double labels. Relaxing $z = 0$ yields the unique edge incident to s at this vertex.

From these observations, it follows that S consists of paths and cycles. We note that some of the edges in S are unbounded. An unbounded edge of S incident to vertex $(\boldsymbol{y}^*, z^*)$ with $z^* > 0$ is called a *ray*. Formally, a ray $\mathcal{R}$ has the form

$$\mathcal{R} = \{(\boldsymbol{y}^*, z^*) + \alpha(\boldsymbol{y}', z') | \alpha \ge 0\},$$

where $(\boldsymbol{y}', z') \ne \boldsymbol{0}$ solves (2) with $\boldsymbol{q} = \boldsymbol{0}$ (the direction vector). Among all rays, one is special. Observe that $\boldsymbol{y} = 0, z \ge |\min_i q_i|$ gives a solution to (2) that forms an unbounded edge of S, known as a *primary ray*. All other rays are called secondary rays. Starting from the primary ray, Lemke's scheme follows a path on the one-skeleton of $\mathcal{P}'$ with a guarantee that it never repeats a vertex. Therefore, either it reaches a vertex with $z = 0$ that is a solution of the original LCP (1) or it ends up on a secondary ray. In the latter case, the algorithm fails to find a solution, and in fact, the problem may not have a solution. Observe that we can replace z with $c_i z$ in (2), where $c_i = 0$ when $q_i > 0$ and $c_i > 0$ when $q_i < 0$ without changing the role of z.

In what follows, for simplicity, we use the shorthand notation of

$$(A\boldsymbol{y})_i \le q_i \perp y_i$$

to represent $\{(A\boldsymbol{y})_i \le q_i;\; y_i \ge 0;\; y_i(A\boldsymbol{y} - \boldsymbol{q})_i = 0\}$ when defining LCPs.

3. LCP Formulation

In this section, we derive an LCP formulation that captures all competitive equilibria of a mixed manna under SPLC utility functions defined in Section 2.1.

3.1. Identifying Goods and Bads

We begin with an important observation. Examining the first segment of each agent's utility function reveals the sign of the item prices at equilibrium. If there exists an agent $i \in N$ such that $U_{ij1} > 0$, then $p_j \ge 0$. Because $U_{ijk} \ge 0$, $\forall k$ for such a j, i demands an infinite amount of j if $p_j < 0$ and then demand will not meet supply. Therefore, in *any* equilibrium, if there exists an agent i such that $U_{ij1} > 0$, then $p_j \ge 0$. Similarly, if $U_{ij1} \le 0$, $\forall i \in N$, then $p_j \le 0$ as, at any positive price, the demand of j is zero. In view of this, we refer to items with a nonnegative price as goods and items with a nonpositive price as bads. Here, a negative price for a bad implies an agent can *earn* by doing (consuming) the chore.

We can further refine these observations to identify situations in which there exists an equilibrium in which an item's price is zero. For any good j, that is, $p_j \ge 0$, we define the desire for j as

$$\text{desire}_j = \sum_{i \in N} \sum_{k: U_{ijk} > 0} L_{ijk}. \tag{3}$$

In words, desire$_j$ is the maximum possible demand for good j at any price $p_j > 0$. Suppose that desire$_j \leq 1$ and then observe that there exists an equilibrium in which $p_j = 0$ because there is a unit amount of each item. Thus, for any good j with desire$_j \leq 1$, we may set $p_j = 0$; allocate the segments that provide positive utility for agents, that is, $U_{ijk} > 0$; and assign any remaining fraction of the good to any zero utility segments.

Similarly, for any bad j, that is, $p_j \leq 0$, we define the indifference to j as

$$\text{indifference}_j = \sum_{i \in N} \sum_{(i,j,1): U_{ij1}=0} L_{ij1}.$$

The indifference to j is the maximum amount of j that can be assigned without causing any agent to lose utility. If indifference$_j \geq 1$, then observe that there exists an equilibrium in which $p_j = 0$, and the item can be allocated among the agents along segments with $U_{ij1} = 0$.

Henceforth, we assume that the desire for every good is more than one and indifference to every bad is less than one. Further, if the indifference to bad j is less than one, then observe that any segment $(i,j,1)$ for which $U_{ij1} = 0$ is fully bought by agent i. Because we can remove these segments by setting $x_{ij1} := L_{ij1}$, for simplicity, we assume that $U_{ijk} < 0$ for all segments (i, j, k) of bad j.

Remark 1. We note that our results generalize to the setting in which an agent's utility function for an item is arbitrary piecewise linear concave, subsuming the studied case in which an item is either a good or a bad for an agent. For this, first compute the desire for an item j as desire$_j = \sum_{i \in N} \sum_{k: U_{ijk} \geq 0} L_{ijk}$. The discussion implies that, if desire$_j > 1$, we can consider it as a good by setting $U_{ijk} = \max\{0, U_{ijk}\}$ for every (i, j, k). If desire$_j \leq 1$, we can consider it as a bad by setting $x_{ijk} = L_{ijk}$ for every (i, j, k) for which $U_{ijk} \geq 0$. For the boundary case in which desire$_j = 1$, we also set its price p_j to zero.

Note that spending on bads "costs" a negative amount of money because the price is negative for any bad. The natural economic interpretation is as follows. Suppose agent i accepts some portion of bad j the agent dislikes. As the price of j is negative, this decreases the agent's overall spending. Equivalently, the agent increases the agent's budget by accepting responsibility for handling some universally disliked chore in order to spend more on goods the agent enjoys. Thus, the negative spending on bads can be viewed as receiving payment on some chore j to increase the agent's budget.

3.2. Characterizing Optimal Bundles

At any prices, for each u_{ij} function, clearly segment $k \geq 1$ is more attractive to agent i than any later segment $k' > k$ because of the concavity of u_{ij}. Therefore, even if agent i is allowed to buy "segments" of u_{ij}, the agent buys them in increasing order. Formally, given a vector of prices $\boldsymbol{p}$, an optimal bundle of items for agent i, that is, the bundle that maximizes the agent's utility subject to the *budget constraint*, solves the following LP:

$$\max \sum_{j,k} U_{ijk} x_{ijk} \quad \text{s.t.} \sum_{j,k} x_{ijk} p_j \leq \sum_j W_{ij} p_j; \quad 0 \leq x_{ijk} \leq L_{ijk}, \quad \forall (i,j,k),$$

where x_{ijk} is the fraction of item j allocated to agent i on the kth segment of u_{ij}. However, we require a more explicit characterization for later analysis.

For any good j, define the bang per buck (*bpb*) of agent i on segment (j, k) as

$$bpb_{ijk} = \frac{U_{ijk}}{p_j}.$$

Note that, bpb_{ijk} is the utility gained per unit spending on the kth segment of good j. Similarly, for any bad j, define the pain per buck (*ppb*) of agent i on segment (j, k) as

$$ppb_{ijk} = \frac{U_{ijk}}{p_j}.$$

Note that, for a bad j, because $p_j < 0$ and $U_{ijk} \leq 0$, we have $ppb_{ijk} \geq 0$, and it is the disutility per unit earning on the kth segment of bad j.

Intuitively, optimal bundles for any agent consist of segments with maximum *bpb* for goods, which yield the highest utility per unit spending, and minimum *ppb* for bads, which minimizes disutility per unit spending. This can be easily verified through Karush–Kuhn–Tucker conditions on the preceding LP. Given prices $\boldsymbol{p}$, these segments can be computed as follows. Sort agent i's segments for goods in decreasing order of bpb_{ijk} and increasing order of ppb_{ijk} for bads. Define the equivalence classes (partition) $G_1, \ldots, G_l$ for goods with equal bpb_{ijk} and

$B_1, \ldots, B_{l'}$ with equal ppb_{ijk} for bads. Observe that each segment in G_d adds an equal amount of utility per unit spending, whereas each segment in $B_{d'}$ adds an equal amount of disutility per unit earning. Obviously, agent i demands G_ds and $B_{d'}$s in the increasing order to maximize the agent's utility subject to the budget constraint. By abuse of notation, we use $bpb(G_d)$ (respectively, $ppb(B_{d'})$) to denote the bpb (ppb) of the segments in equivalence class G_d ($B_{d'}$).

Because an agent's utility decreases by consuming chores, the agent consumes one only if the agent needs the money earned to either satisfy the agent's budget constraint (pay for the chores the agent owns) or use it to buy goods that (over)compensate for the disutility. Therefore, for agent i, if $\sum_{j\in M} W_{ij}p_j \geq 0$, then the agent consumes a segment from $B_{d'}$ only if there exists a G_d such that $bpb(G_d) \geq ppb(B_{d'})$. If the latter inequality is strict, then i chooses to accept as much bads as possible from $B_{d'}$ to buy goods from G_d.

Suppose agent i stops buying goods and bads at equivalence classes G_d and $B_{d'}$, respectively; G_d ($B_{d'}$) is the first partition that is not fully consumed. We note that, if $\sum_{j\in M} W_{ij}p_j < 0$, then agent i may consume only chores to earn the desired money. For all $k < d$ and $k' < d'$, we call the segments of equivalence classes G_k and $B_{k'}$ *forced*. All the segments of equivalence classes G_d and $B_{d'}$ are called *flexible*. And, for all $k > d$ and $k' > d'$, we call the segments of G_k and $B_{k'}$ *undesirable*. For all agents, $ppb \geq bpb$ in their flexible partition.

3.3. LCP Formulation for All Bads

In this section, we derive an LCP formulation to capture competitive equilibria for the case when mixed manna contains only bads, that is, $U_{ijk} \leq 0$, $\forall(i,j,k)$. We build on the approaches of Eaves [22] and Garg et al. [31] for only goods. Our task consists of two steps. First, we need to design constraints to ensure that the market clears (i.e., all bads are fully allocated and each agent earns exactly the required budget). Second, we need to ensure agents earn their budget on optimal bundles of bads.

The first problem, market clearing, is straightforward and does not even require complementarity. Note that the LCP formulation requires nonnegative variables. However, prices and spending on bads are negative. Therefore, we create nonnegative variables p_j for all $j \in M$, and f_{ijk} for all segments (i, j, k). We use $(-p_j)$ as the price of bad $j \in M$ and $(-f_{ijk})$ as the amount agent i spends on segment (i, j, k). We also let $D_{ijk} = |U_{ijk}|$ denote i's disutility on segment (i, j, k). Also, for each agent i, we introduce a variable r_i to capture the reciprocal of the pain per buck of agent i's flexible partition.

Let $\perp$ denote a complementarity constraint between an inequality and a variable (e.g., $\sum_j W_{ij}p_j \leq \sum_{j,k} f_{ijk} \perp r_i$ is a shorthand for $\sum_j W_{ij}p_j \leq \sum_{j,k} f_{ijk}$; $r_i \geq 0$; $r_i(\sum_j W_{ij}p_j - f_{ijk}) = 0$). We ensure market clearing with the following constraints, in which each variable is paired with a constraint by complementarity conditions to yield a standard LCP formulation:

$$\forall i \in N: \sum_j W_{ij}p_j \leq \sum_{j,k} f_{ijk} \perp r_i, \tag{3a}$$

$$\forall j \in M: \sum_{j,k} f_{ijk} \leq p_j \perp p_j. \tag{3b}$$

Lemma 1. *If $\boldsymbol{p}^*$ is an equilibrium price vector, then $\exists \boldsymbol{f} = (f_{ijk}s)$ such that (3a) and (3b) hold. Further, if $\boldsymbol{p}$ and $\boldsymbol{f}$ satisfy (3a) and (3b) and $\boldsymbol{p} > 0$, then the market clears.*

Proof. Let $\boldsymbol{p}^*$ be an equilibrium price vector and set $\boldsymbol{p} = |\boldsymbol{p}^*|$. Let $\boldsymbol{x}^*$ be an equilibrium allocation at $\boldsymbol{p}^*$. For each agent i and each bad j, we distribute x^*_{ij} among individual segments by filling, starting from the first segment until all of x^*_{ij} is used, that is,

$$x^*_{ijk} = \min\left(\max\left(x^*_{ij} - \sum_{k'<k} L_{ijk}, 0\right), L_{ijk}\right). \tag{4}$$

The market clearing condition ensures that setting $f_{ijk} = x^*_{ijk}p_j$ together with $\boldsymbol{p}$ satisfies (3a) and (3b). For the second claim, suppose $\boldsymbol{p}, \boldsymbol{f}$ satisfy (3a) and (3b) and $\boldsymbol{p} > \mathbf{0}$. Summing (3a) over all $i \in N$ and (3b) over all $j \in M$ gives

$$\sum_j p_j = \sum_{i,j} W_{ij}p_j \leq \sum_{i,j,k} f_{ijk} \leq \sum_j p_j,$$

where the first equality uses the fact that there is a unit amount of each bad, that is, $\sum_i W_{ij} = 1$, $\forall j \in M$. It follows from the nonnegativity of all variables that all Constraints (3a) and (3b) hold with equality. Therefore, setting $x_{ijk} = f_{ijk}/p_j$ ensures that the market clears. □

Next, we design constraints to ensure agents purchase optimal bundles of bads. Recall the characterization of optimal bundles from Section 3. Let (i, j, k) be a segment of agent i's flexible partition. We want variable r_i to satisfy

$$ppb_{ijk} = \frac{1}{r_i} = \frac{D_{ijk}}{p_j} > 0. \tag{5}$$

For any forced segment (i, j', k'), we have $ppb_{ij'k'} < ppb_{ijk}$. We compensate for this by adding another variable $s_{ij'k'} \geq 0$ for each segment (i, j', k') of i's utility function. We want $s_{ijk} > 0$ for any forced segment and $s_{ijk} = 0$ otherwise. The new variables can be interpreted as discount prices for each segment of i's utility function. This leads to the following constraints and complementarity conditions:

$$\forall (i,j,k): \; p_j - s_{ijk} \leq D_{ijk} r_i \perp f_{ijk}, \tag{3c}$$

$$\forall (i,j,k): \; f_{ijk} \leq L_{ijk} p_j \perp s_{ijk}. \tag{3d}$$

We refer to each constraint by the equation number and the corresponding complementarity condition by the equation number prime. Note that complementarity Condition (3d′) ensures that forced segments are fully purchased. Let us denote the LCP defined by Constraints (3a)–(3d) as LCP (3). The next lemma shows that LCP (3) captures all competitive equilibria.

Lemma 2. *Any competitive equilibrium gives a solution to LCP* (3).

Proof. Let $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ be a competitive equilibrium. Define $\boldsymbol{p}$ and $\boldsymbol{f}$ as in Lemma 1 and set r_i according to (5) for any segment (i, j, k) of i's flexible partition. Note that $r_i > 0$ because $0 < D_{ij1} < D_{ij2} < \ldots$ for all bads $j \in M$. By Lemma 1, (3a) and (3b) hold with equality because $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ clears the market.

We set variables s_{ijk} as follows: if (i, j, k) is undesirable or flexible, set $s_{ijk} = 0$. If (i, j, k) is a forced segment, set s_{ijk} to satisfy

$$\frac{1}{r_i} = \frac{D_{ijk}}{p_j - s_{ijk}} \Rightarrow s_{ijk} = p_j - D_{ijk} r_i.$$

Note that $s_{ijk} \geq 0$ because $D_{ijk} > 0$ and $D_{ijk}/p_j \leq 1/r_i$ for forced segments. It can be easily verified that, in each case (segment is forced, flexible, or undesirable), the Constraints (3c) and (3d), and corresponding complementarity conditions (3c′) and (3d′) are satisfied. □

LCP (3) suffers from a serious problem. The vector corresponding to the vector $\boldsymbol{q}$ in (1) representation contains all zeros, meaning that it admits the trivial solution $\boldsymbol{p} = \boldsymbol{f} = \boldsymbol{r} = \boldsymbol{s} = \boldsymbol{0}$. We address this issue by a change of variables. For any equilibrium price vector $\boldsymbol{p}^*$, there exists a largest price (in magnitude) $P = \max_j |p_j^*|$. Because equilibrium prices are scale invariant, we can assume that P is a positive constant. Changing variables to define prices relative to P makes $-(P - p_j)$ the price of bad $j \in M$. Observe that bounding the maximum price (in absolute value) also bounds each agent's ppb in the agent's flexible partition, that is, $1/ppb \leq P/D_{min}$, $\forall i \in N$, where $D_{min} = \min_{i,j,k: D_{ijk} > 0} D_{ijk}$. Let a constant $R > (P/D_{min})$ and replace r_i with $(R - r_i)$. That is, we want $1/(R - r_i) = ppb_i$ for i's flexible partition. Substituting the new variables (i.e., p_j with $(P - p_j)$ and r_i with $(R - r_i)$) into LCP (3) yields

$$\forall i \in N: -\sum_j W_{ij} p_j - \sum_{j,k} f_{ijk} \leq -P \sum_j W_{ij} \perp r_i, \tag{6a}$$

$$\forall j \in M: \sum_{i,k} f_{ijk} + p_j \leq P \perp p_j, \tag{6b}$$

$$\forall (i,j,k): D_{ijk} r_i - p_j - s_{ijk} \leq D_{ijk} R - P \perp f_{ijk}, \tag{6c}$$

$$\forall (i,j,k): f_{ijk} + L_{ijk} p_j \leq L_{ijk} P \perp s_{ijk}. \tag{6d}$$

LCP (6) still allows one noncompetitive equilibrium. Observe that setting $p_j = P$, $\forall j \in M$, $r_i = R$, $\forall i \in N$ and all other variables $(\boldsymbol{f}, \boldsymbol{s}) = \boldsymbol{0}$ solves LCP (6), but this solution is not a competitive equilibrium. Rather, this degenerate "equilibrium" proposes to make the price of each bad zero because the price of bad j is $-(P - p_j)$. In turn, this makes each agent's budget equal to zero and prevents the agent from earning on anything. Ultimately, this leaves all bads unallocated, and the market doesn't truly clear. We call this the degenerate solution. We show in Section 4 that the algorithm never reaches this solution. Assuming $p_j < P$, $\forall j \in M$ and $r_i < R$, $\forall i \in N$, it is straightforward to verify that Lemma 2 still holds.

Lemma 3. *In any solution to LCP (6) with $p_j < P,\ \forall j \in M$ and $r_i < R,\ \forall i \in N$, each agent receives an optimal bundle of bads.*

Proof. Recall that $D_{ijk} = |U_{ijk}| > 0,\ \forall (i,j,k)$ because, for each agent, the agent's utility for each bad is a concave, decreasing function, that is, $0 < D_{ij1} < D_{ij2} < \ldots$. Because of the scale invariance of competitive equilibria, we may pick any maximum price (in absolute value) P. Given the choice of P, we selected R such that $R > P/D_{min}$. This ensures $D_{ijk}R - P > 0,\ \forall (i,j,k)$, which makes the right-hand side of (6c) positive for all segments (i,j,k). This implies that $r_i > 0,\ \forall i \in N$; otherwise, (6c) is a strict inequality. In turn, (6c′) forces $f_{ijk} = 0,\ \forall (i,j,k)$. Then, for each $j \in M$, $p_j < P$ implies (6b) is strict, and therefore, we have $p_j = 0$, which violates Inequality (6a).

Let (i, j, k) be a segment with the highest *ppb* that agent i spends on in the solution to LCP (6). Define σ_i as the reciprocal of the pain per buck of this segment:

$$\sigma_i = \frac{P - p_j}{D_{ijk}} = \frac{1}{ppb_{ijk}}.$$

Observe that $\sigma_i > 0$ because $p_j < P$ and $D_{ijk} > 0$ for all segments of all bads.

We want to show that $(R - r_i) \le \sigma_i$. Because agent i spends on segment (i, j, k), that is, $f_{ijk} > 0$, complementarity Condition (6c′) requires that Constraint (6c) holds with equality. Because $s_{ijk} \ge 0$, this yields

$$D_{ijk}(R - r_i) = P - p_j - s_{ijk} \le P - p_j = D_{ijk}\sigma_i. \tag{7}$$

Thus, $(R - r_i) \le \sigma_i$ because $D_{ijk} > 0$.

Let Q_i denote all segments of i's utility function with $ppb = 1/\sigma_i$ and call this the flexible partition. Similarly, let the forced partition be all segments with strictly lower *ppb* than $1/\sigma_i$ and let the undesirable partition be all segments with strictly higher *ppb* than $1/\sigma_i$. We show that these segments correspond to the forced, flexible, and undesirable partitions described in Section 3.

Observe that undesirable partitions are unallocated by construction because we selected σ_i based on the segment receiving a positive allocation with the highest *ppb*. Now, consider any segment (i, j, k) in agent i's forced partition. We have

$$\frac{D_{ijk}}{P - p_j} < \frac{1}{\sigma_i} \Rightarrow P - p_j > D_{ijk}\sigma_i \ge D_{ijk}(R - r_i).$$

Hence, to satisfy (6c), it must be that $s_{ijk} > 0$. Therefore, (6d) must hold with equality to satisfy (6d′). That is, the segment is fully allocated.

Finally, let $(i,j,k) \in Q_i$. If $(R - r_i) < \sigma_i$, then all segments of this partition are also fully allocated by the similar argument as earlier. In other words, the agent exhausts the agent's budget when the agent is done consuming Q_i as the last partition. It follows from the characterization in Section 3 that each agent receives an optimal bundle of bads. □

Theorem 3. *The solutions to LCP (6) with $p_j < P,\ \forall j \in M$ and $r_i < R,\ \forall i \in N$ exactly captures all competitive equilibria (up to scaling).*

Proof. By Lemma 1 at prices $p_j^* = -(P - p_j)$ for all j the market clears. And, by Lemma 3, each agent receives an optimal bundle of bads in any solution to LCP (6), that is, it is a competitive equilibrium. Further, up to change of variables from LCP (3) to LCP (6), Lemma 2 shows that every competitive equilibrium yields a solution to LCP (6). □

3.4. LCP Formulation for Mixed Manna

We now extend the LCP formulation to the general mixed manna case. Given the known LCP formulation for SPLC utilities for good manna from Garg et al. [31] and LCP (6) for all bads, a natural question is can we simply combine an LCP for goods and an LCP for bads to obtain an LCP for mixed manna? Note that this treats the mixed manna case as two separate subproblems: one for goods and one for bads. Such a formulation requires separate budget constraints for goods and bads; that is, each agent's spending on goods (bads) is at least as much as the agent's earnings on goods (bads), similar to Constraint (3a). However, a simple example illustrates that, in general, this is not possible.

Example 1. Consider an instance with two agents A and B and two items 1 and 2. Agents' utilities are as follows: $u_A(x_A) = x_{A1} - 2x_{A2}$, and $u_B(x_B) = x_{B1} - 3x_{B2}$. Assume each agent brings an equal amount of each item, that is, $W_{A1} = W_{A2} = W_{B1} = W_{B2} = 0.5$.

There are a few important things to note. Both agents like item 1, so it is a good and $p_1 > 0$, and because both agents dislike item 2, it is a bad and $p_2 < 0$. Clearly, both agents must purchase some of bad 2 at equilibrium. A portion of item 1 cannot be purchased by both agents because optimal bundles require *bpb* = *ppb*. Thus, if both agents purchase some of item 1, then $u_{A1}/p_1 = u_{A2}/p_2$ or $p_2 = -2p_1$, but we also have the requirement $p_2 = -3p_1$, a contradiction. Therefore, only one agent purchases good 1. One can verify that prices $p_1 = 2$ and $p_2 = -4$ along with allocation $x_{A1} = 1$, $x_{A2} = 3/4$, $x_{B1} = 0$, and $x_{B2} = 1/4$ are an equilibrium in which each agents' initial budget is set to –1. Note that agent 1's total spending on the good (i.e., item 1) is two, and agent 2's spending on the good is zero. However, the total value of good in each agent's initial bundle is one. Thus, neither agent's spending on the good equals the value of good in the agent's initial endowment.

3.4.1. Basic Formulation. Similar to Section 3.3, we start by designing an LCP whose solutions capture competitive equilibria. This requires that the market clears and agents purchase optimal bundles of goods and bads. Although specialized to the case of all bads, the derivation of LCP (6) in Section 3.3 provides the basic framework needed to handle the general mixed manna setting. As discussed in Section 3, we can identify which items are goods and which are bads by examining the sign of the utility for the first segment of each agent. For every $U_{ij1} \le 0$ for a good j, we set $f_{ijk} := 0$ at the beginning itself and do not introduce the corresponding variables in our formulation. Note that, when dealing with mixed manna, we assume strong connectivity (see Conditions 1 and 2).

For clarity, we first write all complementarity conditions with a minimal change of variables. Let M^- and M^+ denote the set of bads and goods, respectively. For all $j \in M^-$, prices, spending, and utilities are negative. We introduce nonnegative variables p_j and f_{ijk} for all $j \in M$. We interpret p_j as the price of good $j \in M^+$ and $(-p_j)$ as the price of bad $j \in M^-$. Similarly, f_{ijk} gives agent i's spending on segment (i, j, k) for good j, whereas $-f_{ijk}$ gives i's spending on segment (i, j, k) of bad $j \in M^-$. We ensure market clearing with the following complementarity conditions:

$$\forall i \in N: \sum_{k, j \in M^+} f_{ijk} - \sum_{k, j \in M^-} f_{ijk} \le \sum_{j \in M^+} W_{ij} p_j - \sum_{j \in M^-} W_{ij} p_j \perp r_i, \tag{8a}$$

$$\forall j \in M^-: \sum_{i,k} f_{ijk} \le p_j \perp p_j, \tag{8b}$$

$$\forall j \in M^+: p_j \le \sum_{i,k} f_{ijk} \perp p_j. \tag{8c}$$

Note that we treat the spending constraints for bads (8b) and goods (8c) differently. Further, if all items are bads, then we recover (3a) and (3b).

Lemma 4. *If $\boldsymbol{p}^*$ is an equilibrium price vector, then $\exists \boldsymbol{f}$ such that $(\boldsymbol{p}, \boldsymbol{f})$ satisfies (8a)–(8c), where $\boldsymbol{p} = |\boldsymbol{p}^*|$. Further, if $\boldsymbol{p}$ and $\boldsymbol{f}$ satisfy (8a)–(8c) and $\boldsymbol{p} > 0$, then the market clears.*

Proof. Let $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ be an equilibrium. Set $p_j = |p_j^*|,\ \forall j \in M^-$, and $p_j = p_j^*,\ \forall j \in M^+$. For each agent i and each item j, we distribute x_{ij}^* among individual segments by filling starting from the first segment until all of x_{ij}^* is used, according to (4). The market clearing conditions for an equilibrium ensure that setting $f_{ijk} = x_{ijk}^* p_j$ together with $\boldsymbol{p}$ satisfies (8a)–(8c).

For the other case, suppose $\boldsymbol{p}, \boldsymbol{f}$ satisfy (8a)–(8c) and $\boldsymbol{p} > 0$. Summing (8a) over all $i \in N$, (8b) over all $j \in M^-$, and (8c) over all $j \in M^+$ gives

$$\sum_{j \in M^+} p_j - \sum_{j \in M^-} p_j = \sum_{i,k,j \in M^+} f_{ijk} - \sum_{i,k,j \in M^-} f_{ijk} \le \sum_{i,j \in M^+} W_{ij} p_j - \sum_{i,j \in M^-} W_{ij} p_j = \sum_{j \in M^+} p_j - \sum_{j \in M^-} p_j,$$

because there is a unit amount of each item, that is, $\sum_i W_{ij} = 1$. Because all variables are nonnegative, it follows that (8a)–(8c) hold with equality. Therefore, setting $x_{ijk} = f_{ijk}/p_j$ ensures that the market clears. □

The next step is to make sure agents purchase optimal bundles. Let (i, j, k) be a segment of agent i's flexible partition. We use variable r_i to satisfy

$$\frac{1}{r_i} = \frac{U_{ijk}}{p_j} \quad \text{if } j \in M^+, \text{ and } \quad \frac{1}{r_i} = \frac{D_{ijk}}{p_j} \quad \text{if } j \in M^-, \tag{9}$$

where $D_{ijk} = |U_{ijk}|$ for bad j. Recall that forced segments of goods and bads correspond to slightly different conditions. For any forced segment (i, j, k'), we have $ppb_{ijk'} < ppb_{ijk}$. For any forced segment (i, j, k'), we have $bpb_{ijk'} > bpb_{ijk}$. Again, we compensate for this by introducing a variable $s_{ijk} \ge 0$ into each segment (i, j, k) of i's utility function, leading to the following complementarity conditions:

$$\forall j \in M^-,\ \forall (i,j,k):\ p_j - s_{ijk} \le D_{ijk} r_i \perp f_{ijk}, \tag{8d}$$

$$\forall j \in M^+,\ \forall (i,j,k):\ U_{ijk} r_i \le p_j + s_{ijk} \perp f_{ijk}, \tag{8e}$$

$$\forall (i,j,k):\ f_{ijk} \le L_{ijk} p_j \perp s_{ijk}. \tag{8f}$$

Observe that, if all items are bads, then we recover (3c) and (3d). Let us denote the LCP defined by Constraints (8a)–(8f) as LCP (8).

Lemma 5. *Any competitive equilibrium of mixed manna gives a solution to* LCP (8).

Proof. Let $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ be a competitive equilibrium. In the LCP set $p_j = |p_j^*|,\ \forall j \in M$ and $\boldsymbol{f}$ as done in the proof of Lemma 4. By Lemma 4, (8a)–(8c) hold with equality because $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ clears the market. Therefore, so do (8a′), (8b′), and (8c′). For each agent i, if i purchases any goods, then set r_i according to (9) for any flexible segment of goods. Otherwise, set r_i according to (9) for any flexible segment of bads. In either case, $r_i > 0$ because i is nonsatiated for some good j, that is, $U_{ijk} > 0$ and $0 < D_{ij1} < D_{ij2} < \ldots$ for all bads $j \in M^-$.

We set variables s_{ijk} as follows: if (i, j, k) is undesirable or flexible set $s_{ijk} = 0$ whether j is a good or a bad. Recall that, for a forced segment (i, j, k), if $j \in M^+$, then $\frac{1}{r_i} < \frac{U_{ijk}}{p_j}$, and if $j \in M^-$, then $\frac{1}{r_i} > \frac{U_{ijk}}{p_j}$. Using this, set its s_{ijk} to satisfy

$$\frac{1}{r_i} = \frac{U_{ijk}}{p_j + s_{ijk}},\ \text{if } j \in M^+, \quad \text{or} \quad \frac{1}{r_i} = \frac{D_{ijk}}{p_j - s_{ijk}},\ \text{if } j \in M^-.$$

It is easy to verify that, in each case (segment is forced, flexible, or undesirable), Constraints (8d)–(8f) are satisfied as well as the corresponding complementarity conditions (8d′)–(8f′). □

Similar to the case of all bads, LCP (8) admits solutions that are not competitive equilibria, for example, the trivial solution $\boldsymbol{p} = \boldsymbol{f} = \boldsymbol{r} = \boldsymbol{s} = \boldsymbol{0}$. We use the same change of variables as before. We fix a maximum price (in absolute value) P and define the relative prices: $(P - p_j)$ for all goods $j \in M^+$ and $-(P - p_j)$ for all bads $j \in M^-$. This bounds each agent's *ppb* or *bpb* in the agent's flexible partition, that is, $1/bpb, 1/ppb \le P/U_{min}$, where $U_{min} = \min_{i,j,k: U_{ijk} \neq 0} |U_{ijk}|$. Using this, we define a constant $R > P/U_{min}$ and replace r_i with $(R - r_i)$. That is, we want $\frac{1}{(R - r_i)}$ to represent *ppb* and *bpb* of agent i's flexible partition. Substituting p_j with $(P - p_j)$ for each $j \in M$ and r_i with $(R - r_i)$ for each agent $i \in N$ into LCP (8) yields

$$\forall i \in N: \sum_{j \in M^+} W_{ij} p_j - \sum_{j \in M^-} W_{ij} p_j + \sum_{k, j \in M^+} f_{ijk} - \sum_{k, j \in M^-} f_{ijk} \le P\left(\sum_{j \in M^+} W_{ij} - \sum_{j \in M^-} W_{ij}\right) \perp r_i, \tag{10a}$$

$$\forall j \in M^-: \sum_{i,k} f_{ijk} + p_j \le P \perp p_j, \tag{10b}$$

$$\forall j \in M^+: -\sum_{i,k} f_{ijk} - p_j \le -P \perp p_j, \tag{10c}$$

$$\forall j \in M^-,\ \forall i,k: D_{ijk} r_i - p_j - s_{ijk} \le D_{ijk} R - P \perp f_{ijk}, \tag{10d}$$

$$\forall j \in M^+,\ \forall i,k: -U_{ijk} r_i + p_j - s_{ijk} \le P - U_{ijk} R \perp f_{ijk}, \tag{10e}$$

$$\forall (i,j,k): f_{ijk} + L_{ijk} p_j \le L_{ijk} P \perp s_{ijk}. \tag{10f}$$

In the case of all bads, LCP (10) is equivalent to LCP (6) from Section 3.3.

Similar to LCP (6), LCP (10) still allows (at least) one noncompetitive equilibrium. By setting $p_j = P$, $\forall j \in M$, $r_i = R$, $\forall i \in N$, and all other variables $(\boldsymbol{f}, \boldsymbol{s}) = \boldsymbol{0}$, we get a solution to LCP (10). However, this solution does not correspond to a competitive equilibrium, but rather a degenerate solution in which all prices are zero and no items are allocated. We show in Section 4.1 that the algorithm never reaches this degenerate solution. Assuming $p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$, it is straightforward to verify that Lemmas 4 and 5 still hold.

Lemma 6. *In any solution to* LCP (10) *with $p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$, all agents receive an optimal bundle and market clears with respect to prices $\boldsymbol{p}^*$, where $p_j^* = (P - p_j)$ for all goods $j \in M^+$ and $p_j^* = -(P - p_j)$ for all chores $j \in M^-$.*

Proof. Given a solution of LCP (10), define $p_j^* = (P - p_j),\ \forall j \in M^+$ and $p_j^* = -(P - p_j),\ \forall j \in M^-$ and allocation $x_{ij}^* = \sum_k f_{ijk}/p_j^*,\ \forall (i,j)$. We want to show that $(\boldsymbol{p}^*, \boldsymbol{x}^*)$ gives a competitive equilibrium. It is easy to show that the market clears at $(\boldsymbol{p}^*, \boldsymbol{x}^*)$ using (10a)–(10c) via a similar argument as in Lemma 4. Next, we show that every agent receives an optimal bundle as per $\boldsymbol{x}^*$ at prices $\boldsymbol{p}^*$.

Recall that we picked P and R such that $\min_{j \in M^-, i,k} D_{ijk} R - P > 0$, and $P - \min_{j \in M^+, i.k} U_{ijk} R < 0$. Whereas similar to the proof of Lemma 3, we now rely on the assumption that each agent i is nonsatiated for some good j; that is, the final segment (i, j, k) of good j satisfies $U_{ijk} > 0$. Notice that because $D_{ijk} > 0,\ \forall i,k$ for any bad j, and the preceding assumption on goods implies that $r_i > 0,\ \forall i \in N$. Consider two cases: an agent purchases some bads or only goods. In the first case, if $r_i = 0$, then (10d) is a strict inequality. Then, (10d′) requires $f_{ijk} = 0,\ \forall k,\ \forall j \in M^-$, contradicting the assumption that i purchases some bads. Similarly, in the second case, if $r_i = 0$, then (10e) cannot hold for the nonsatiated segment with infinite length.

Here, we diverge from the case of all bads, depending on whether an agent purchases any goods (or bads). Consider any agent i. There are three cases, i purchases (a) only goods, (b) only bads, or (c) goods and bads. We focus on the last case as it is the most complicated. The first two cases can be handled in a similar manner. Let (i, j, k) be the segment of goods with the lowest bang per buck on which agent i spends, that is, $f_{ijk} > 0$. Define ν_i as the reciprocal *bpb* of this segment

$$\nu_i = \frac{P - p_j}{U_{ijk}}.$$

Note that $0 < \nu_i < \infty$ because $p_j < P$, and each agent is nonsatiated for some good j. Similarly, let (i, j', k') be the segment of bads with the highest pain per buck, *ppb*, on which i spends, and define σ_i as in Lemma 3. We want to show that $\nu_i \le (R - r_i) \le \sigma_i$. Therefore, $bpb_{ijk} \ge ppb_{ij'k'}$ for any good j and any bad j' on which i spends. From (7), we have $(R - r_i) \le \sigma_i$. By a similar argument, for the segment (i, j, k) with the lowest bang per buck,

$$U_{ijk}(R - r_i) = (P - p_j) + s_{ijk} \ge (P - p_j) = U_{ijk}\nu_i.$$

Thus, $(R - r_i) \ge \nu_i$.

Let G_i denote the set of segments of goods with $bpb = 1/\nu_i$ and call this the flexible partition of goods. Similarly, let the forced partition of goods be all segments with strictly higher *bpb* than $1/\nu_i$ and let the undesirable partition of goods be all segments with strictly lower *bpb* than $1/\nu_i$. Define the various partitions of bads as forced for *ppb* strictly less than σ_i, undesirable if *ppb* is strictly more than σ_i, and let B_i be the flexible partition for bads in which $ppb = 1/\sigma_i$. As per the optimal bundle characterization described in Section 3, we need to show that f_{ijk}'s are zero for the segments in undesirable partitions, $f_{ijk} = L_{ijk}(P - p_j)$ for the segments in forced partitions, and $0 \le f_{ijk} \le L_{ijk}(P - p_j)$ for segments in G_i and B_i.

Observe that undesirable goods (bads) are unallocated by construction because we selected ν_i (σ_i) based on the segment receiving a positive allocation with the lowest *bpb* (highest *ppb*). Consider any segment (i, j, k) in agent i's forced partition, whether a bad or a good. Observe that $s_{ijk} > 0$ in order to satisfy (10d) or (10e). Therefore, (10f') requires that (10f) holds with equality. That is, the segment is fully allocated.

For the flexible partition, if $\nu_i < (R - r_i)$, then for all $(i, j, k) \in G_i$, it must be that $s_{ijk} > 0$, and hence, $f_{ijk} = L_{ijk}(P - p_j)$; otherwise, (i, j, k) could be partially allocated. Similarly, if $(R - r_i) < \sigma_i$, then all the segments in B_i are fully allocated; otherwise, they could be partially allocated. Thus, i only purchases goods with $bpb \ge ppb$, and in the flexible partition of goods and bads, $bpb = ppb$. It follows from the characterization in Section 3 that each agent receives an optimal bundle of bads. □

The next theorem follows using Lemmas 5 and 6 together with the way LCP (10) is constructed from LCP (8).

Theorem 4. *The solutions to LCP (10) with $p_j < P$, $\forall j \in M$, and $r_i < R$, $\forall i \in N$, exactly captures competitive equilibrium of mixed manna (up to scaling).*

Proof. By Lemmas 4 and 6, the market clears and each agent receives an optimal bundle of items in any solution to LCP (10) with $p_j < P$, $\forall j \in M$ and $r_i < R$, $\forall i \in N$; that is, it is a competitive equilibrium. Further, Lemma 5 shows that every competitive equilibrium price yields a solution to LCP (10) with $p_j < P$, $\forall j \in M$ and $r_i < R$, $\forall i \in N$. Therefore, solutions to LCP (10) with $p_j < P$, $\forall j \in M$, and $r_i < R$, $\forall i \in N$, exactly captures competitive equilibria up to scaling. □

3.4.2. Augmented LCP and Nondegeneracy. Observe that LCP (10) has the same form as (1) in Section 2.3. We now give the augmented LCP for this problem. By the choice of P and $R > P/\min_{i,j,k:U_{ijk}\neq 0}|U_{ijk}|$ we have that, for all bads $j \in M^-$, $D_{ijk}R - P > 0$, $\forall i, k$. This makes the right-hand side of (10d) positive for all bads. Standard LCP techniques (Eaves [22], Garg et al. [30, 31]) add variable z only in constraints with a negative right-hand side. We make two changes. First, we include variable z in any constraints with a negative right-hand side and *all* budget constraints (10a). Second, when adding z into spending constraints for goods (10c), we use a coefficient $\delta_j = 1 + 1/(m + j)$, where we use $1 \le j \le |M^+|$ as the index of the jth good $j \in M^+$. This change is necessary to ensure that the polyhedron corresponding to the augmented LCP remains nondegenerate as discussed shortly. Adding z to any constraints with a negative right-hand side and *all* budget constraints (10a) yields

$$\forall i \in N: \sum_{j\in M^+} W_{ij}p_j - \sum_{j\in M^-} W_{ij}p_j + \sum_{k,j\in M^+} f_{ijk} - \sum_{k,j\in M^-} f_{ijk} - z \le P\left(\sum_{j\in M^+} W_{ij} - \sum_{j\in M^-} W_{ij}\right) \perp r_i, \tag{11a}$$

$$\forall j \in M^-: \sum_{i,k} f_{ijk} + p_j \le P \perp p_j, \tag{11b}$$

$$\forall j \in M^+ : -\sum_{i,k} f_{ijk} - p_j - \delta_j z \le -P \perp p_j, \tag{11c}$$

$$\forall j \in M^-,\ \forall i,k : D_{ijk} r_i - p_j - s_{ijk} \le D_{ijk} R - P \perp f_{ijk}, \tag{11d}$$

$$\forall j \in M^+,\ \forall i,k : -U_{ijk} r_i + p_j - s_{ijk} - z \le P - U_{ijk} R \perp f_{ijk}, \tag{11e}$$

$$\forall (i,j,k) : f_{ijk} + L_{ijk} p_j \le L_{ijk} P \perp s_{ijk}, \tag{11f}$$

$$z \ge 0. \tag{11g}$$

Let $\mathcal{P}$ be the polyhedron corresponding to LCP (11). Lemke's algorithm requires nondegeneracy of the polyhedron $\mathcal{P}$; that is, if $\mathcal{P}$ is defined on k variables, then at any d dimensional face of $\mathcal{P}$, exactly $(k - d)$ inequalities hold with equality. In that case, the solutions of LCP (11) are paths and cycles on the one-skeleton of $\mathcal{P}$. However, there is an inherent degeneracy present when $z = 0$, that is, solutions of LCP (10).

Inherent degeneracy in LCP (10): summing (10a) over all $i \in N$ and (10b) over all $j \in M^-$ and (10c) over all $j \in M^+$ yields two identical equations; see the proof of Lemma 4 for details. That is, there is an inherent degeneracy in $\mathcal{P}$.

Clearly, the inherent degeneracy of LCP (10) is still present in $\mathcal{P}$ when $z = 0$. We need to show that no other degeneracies exist. Note that, by using $\delta_j = 1 + 1/(m + j)$ for all goods $j \in M^+$, we preclude the following degeneracy at the primary ray: suppose $\delta_j = 1,\ \forall j \in M^+$ and that $p_j = 0,\ \forall j \in M^+$ and $f_{ijk} = 0$ for all segments (i, j, k) of each good. If $z = P$, then (11c) also holds with equality $\forall j \in M^+$. Therefore, we have double labels for each good $j \in M^+$.

If there is a degenerate vertex $\boldsymbol{v} \in \mathcal{P}$ with $z > 0$, $p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$, then using the extra tight inequalities at $\boldsymbol{v}$ we can derive a *polynomial relation* among the input parameters $\boldsymbol{U} = U_{ijk}$s, $\boldsymbol{W} = W_{ij}$s, and $\boldsymbol{L} = L_{ijk}$s, that is, a multivariate polynomial equation in which each monomial is a product of some of $\boldsymbol{U}, \boldsymbol{W}$, and $\boldsymbol{L}$ with integer coefficients. We show this formally in the following two theorems.

Theorem 5. *If the instance parameters $\boldsymbol{U}$, $\boldsymbol{W}$, and $\boldsymbol{L}$ have no polynomial relation among them, then every vertex of $\mathcal{P}$ with $z > 0, p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$, is nondegenerate.*

Proof. We first show this theorem for the case of all bads, that is, for LCP (11) without (11c) and (11e). Let $S = (\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}, \boldsymbol{s}, z)$ be a vertex solution to LCP (11) with $z > 0$ and $p_j < P,\ \forall j \in M$. For a contradiction, suppose S is degenerate. Then, S has at least two double labels. Let $\mathcal{I}$ be the set of inequalities of LCP (11) that hold with equality at S. Remove all zero variables and their nonnegativity conditions from $\mathcal{I}$ as well as all conditions corresponding to double labels at S. Our goal is to write all nonzero variables as linear functions of z, where the coefficients are in terms of monomials of input parameters. Then, substituting these expressions into the double labels at S yields a polynomial relation among input parameters.

For forced segments, that is, $s_{ijk} > 0$, remove conditions (11d) and (11f) from $\mathcal{I}$ and replace f_{ijk} with $L_{ijk}(P - p_j)$. For undesirable segments, (11f) is a strict inequality and $f_{ijk} = 0$. Thus, $\mathcal{I}$ contains no conditions (11d) or (11f) for undesired segments either.

Now, we may write all nonzero variables as linear functions of z. All remaining f_{ijk} correspond to spending in flexible segments. Clearly, for each agent I and each bad j, only one such segment exists. To simplify notation, we relabel f_{ijk} for these flexible segments as f_{ij} and the corresponding D_{ijk} as D_{ij}.

Let $\mathcal{E}$ be the set of (i, j) pairs such that agent i has a flexible segment for bad j, that is, when condition (11d) holds with $s_{ijk} = 0$. Then,

$$D_{ij} r_i - p_j = D_{ij} R - P. \tag{12}$$

By considering the pairs of $\mathcal{E}$ as edges between N and M, we obtain a bipartite graph, say G. Note that G is acyclic; otherwise, we obtain a polynomial relation among D_{ij}'s using (12) along the cycles to eliminate the r_i's and p_j's.

Let H be a connected component of G. We pick a representative bad for H. If there is an undersold bad, that is, (11b) is a strict inequality, then we pick this item, say b. Observe that, for any bad $j \in H$, we may write $P - p_j = \frac{\phi_1(D)}{\phi_2(D)}(P - p_b)$, where $\phi_1(D)$ and $\phi_2(D)$ are monomials in terms of D_{ij}s. Similarly, we may write $R - r_i$ in terms of monomials of D_{ij}s. Now, because (11b) is a strict inequality for bad b, the complementary condition (11b′) requires $p_b = 0$. In addition, no other bad j can be undersold in H; otherwise, these steps yield a polynomial relation among D_{ij}s.

Suppose that, for component H, the representative bad b is not undersold, that is, (11b) holds with equality. Consider any leaf node v_0 of H and remove the edge incident to it in H, say (v_0, v_1), to create H'. Let H' be rooted at v_1. Starting from leaves of H' and working toward the root v_1, we can use market clearing conditions (11b) and (11a) for bads and agents, respectively, to write all f_{ij}'s for edges in H' as linear functions of z and the representative prices obtained in the first step. Market clearing conditions give two different expressions for f_{ij} on the missing edge (v_0, v_1). Thus, yielding a linear relation among the representative prices and z. This relation is nontrivial because exactly one of them must contain a W_{ij} that is not present in the other.

If bad b is undersold, then a similar approach using b as the root allows us to write f_{ij}'s as linear functions of representative prices and z. This gives a system of linear equations: $p_b = 0$ if b is undersold, and p_j is a linear function of representative prices and z otherwise. Solving this system, we obtain p_j's as linear functions of z. Substituting these expressions for representative prices in terms of z, we obtain expressions for f_{ij}'s, r_i's, and remaining p_j's. We perform these steps for each connected component of G.

Finally, consider the equalities of G corresponding to double labels that we removed from $\mathcal{I}$. Replace all variables by their linear functions of z. Use one double label to solve for z in terms of input parameters $\boldsymbol{D}$, $\boldsymbol{W}$, and $\boldsymbol{L}$. Substitute this value of z into the other double label to get a polynomial relation among input parameters, a contradiction.

Now, for the general mixed manna case, the proof closely follows the case of all bads. We assume for a contradiction that the vertex is degenerate. Our goal is to write all nonzero variables as linear functions of z, where the coefficients are in terms of monomials of input parameters. Then, substituting these expressions into the double labels at S yields a polynomial relation among input parameters. Notice that we can still follow these steps in the case of all bads to solve for $\boldsymbol{r}$ as well as $\boldsymbol{p}, \boldsymbol{f}$, and $\boldsymbol{s}$ for all bads $j \in M^-$. Thus, it remains to solve for $\boldsymbol{p}, \boldsymbol{f}$, and $\boldsymbol{s}$ for all goods $j \in M^+$. Using similar arguments to the case of all bads, we find expressions for these variables as linear functions of z. Substituting these expressions into the two sets of double labels yields a polynomial relation among input parameters. □

By a similar argument, together with Theorem 4 and the fact that solutions of LCP (11) with $z = 0$ are solutions of LCP (10), we get the following.

Theorem 6. *If the instance parameters $\boldsymbol{U}$, $\boldsymbol{W}$, and $\boldsymbol{L}$ have no polynomial relation among them, then solutions of* LCP (11) *with $z = 0$, $p_j < P$, $\forall j \in M$, and $r_i < R$, $\forall i \in N$, are in one-to-one correspondence with competitive equilibria.*

Proof. To prove the theorem, it suffices to show one-to-one correspondence between solutions of LCP (10) and competitive equilibria. For this, we first show a similar result for the case of all bads, that is, using LCP (6).

Because of scale invariance, it suffices to show this for the set of competitive equilibria, say $\mathcal{E}$, where the minimum price is $-P$. In LCP (6), we represent the equilibrium price of a bad as $p_j^* = -P + p_j$. Therefore, we show a one-to-one correspondence between elements of $\mathcal{E}$ and solutions to LCP (6) with $p_j = 0$ for some bad j. Let $(\boldsymbol{x}^*, \boldsymbol{p}^*) \in \mathcal{E}$. By Theorem 3, any competitive equilibrium $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ yields a solution to LCP (6) using $\boldsymbol{p} = P + \boldsymbol{p}^*$ and $\boldsymbol{f}$, where $f_{ijk} = x^*_{ijk}(P - p_j)$. We show that this choice of $(\boldsymbol{p}, \boldsymbol{f})$ yields exactly one solution to LCP (6).

For a contradiction, suppose not. Then, there exists different choices of $\boldsymbol{r}$ and $\boldsymbol{s}$ that, together with $(\boldsymbol{p}, \boldsymbol{f})$, solve LCP (6). Observe that fixing $\boldsymbol{p}, \boldsymbol{f}$, and $\boldsymbol{s}$ also fixes $\boldsymbol{r}$. Therefore, it must be true that, for some agent, say i, the agent's flexible partition, say Q_i, is fully allocated, that is, $f_{ijk} = L_{ijk}(P - p_j)$, $\forall (j,k) \in Q_i$. Set r_i so that

$$\frac{1}{R - r_i} = \frac{D_{ijk}}{P - p_j},$$

for some segment $(j,k) \in Q_i$ and set $s_{ijk} = 0$, $\forall\,(j,k) \in Q_i$. Set the $\boldsymbol{r}$ and $\boldsymbol{s}$ for all other agents similarly.

Let $C = \sum_{i,j} |u_{ij}|$ be the total number of segments over all agents and items. Observe that there are $n + m + 2C$ variables in LCP (6). Further, the solution described gives at least $n + m + 2C + 2$ inequalities of LCP (6) that hold with equality: market clearing gives (6a) $\forall i \in N$ and (6b) $\forall j \in M$, and optimal bundles satisfy complementarity conditions (6c′) and (6d′). Plus the requirement $p_j = 0$ for some bad. Finally, all segments of agent i's flexible partition Q satisfy both (6d) and $s_{ijk} = 0$. However, nondegeneracy of LCP (6) means at most $n + m + 2C + 1$ inequalities hold with equality at any vertex.

Now, for the general mixed manna case, showing the one-to-one correspondence follows from a nearly identical argument to that of the case of all bads. The only difference is that we must consider the set of equilibria with the maximum magnitude of price equal to P, that is, $p_j = 0$ for some good or some bad. Assuming a bad has price with the maximum magnitude price P, follow the preceding arguments. The other case with a good also follows from a similar argument. □

Remark 2. We can solve a degenerate instance using the standard ways to handle degeneracy in Lemke's scheme, namely, the *lexico-minimum ratio test* (see Chvátal [14], Cottle et al. [16, section 4.9], Savani [45, section 4.3]) to ensure termination in a finite number of steps.

4. Algorithm

In Section 3.4.2, we design augmented LCP (11) that permits use of Lemke's scheme, and we show a one-to-one correspondence between competitive equilibria and solutions to LCP (11) with $z = 0$ and $p_j < P$, $\forall j \in M$ and $r_i < R$, $\forall i \in N$ so long as the polyhedron $\mathcal{P}$ defined by LCP (11) is nondegenerate. Observe that LCP (11) has the following form:

$$A\boldsymbol{y} - \boldsymbol{c}z \le \boldsymbol{q},\ \boldsymbol{y} \ge 0,\ z \ge 0,\ \text{ and } \boldsymbol{y} \cdot (\boldsymbol{q} - A\boldsymbol{y} + \boldsymbol{c}z) = 0,$$

where $\boldsymbol{c}$ is the vector of coefficients for z in LCP (11). For convenience, we use v_k as a shorthand for $(\boldsymbol{q} - A\boldsymbol{y} + \boldsymbol{c}z)_k$. Note that the condition $\boldsymbol{v} \ge 0$ follows from $\boldsymbol{q} - A\boldsymbol{y} + \boldsymbol{c}z \ge 0$. Of course, when $v_k = (\boldsymbol{q} - A\boldsymbol{y} + \boldsymbol{c}z)_k = 0$, the kth constraint holds. Therefore, at any fully labeled vertex solution S of the polyhedron defined by LCP (11), either $v_k = 0$ or $y_k = 0$. At a double label, $v_k = y_k = 0$. Using the notation of Section 2.3, we let $\boldsymbol{y} = (\boldsymbol{p}, \boldsymbol{f}, \boldsymbol{r}, \boldsymbol{s})$ be a vertex solution to LCP (11).

Recall from Section 2.3 that Lemke's algorithm explores a certain path of the one-skeleton of $\mathcal{P}$, traveling from vertex solution to vertex solution along the edges of $\mathcal{P}$. Note that we chose R such that $R > P/\min_{i,j,k}|U_{ijk}|$, which ensures that the right-hand side of (11d) is positive, that is, $D_{ijk}R - P > 0$, for all segments (i, j, k), $\forall i, k$, $\forall j \in M^-$, and that the right-hand side of (11e) is negative, that is, $P - U_{ijk}R < 0$, for all segments (i, j, k), $\forall i, k$, $\forall j \in M^+$. Further, for sufficiently large R, we have $\min_i P(\sum_{j \in M^+} W_{ij} - \sum_{j \in M^-} W_{ij}) > P - \max_{j \in M^+, i, k} U_{ijk}R$. Then, we get the primary ray (initial solution) S_0 by setting

$$S_0 = \left\{\boldsymbol{y}_0 = \boldsymbol{0},\ z = \max_{j \in M^+, i, k} U_{ijk}R - P,\ \boldsymbol{v}_0 = \boldsymbol{q} + \boldsymbol{c}z\right\}.$$

Clearly, this initialization gives the unique double label $y_{(ijk)^*} = v_{(ijk)^*} = 0$ for $(i, j, k)^* = \arg\max_{(j \in M^+, i, k)} U_{ijk}R - P$.

Algorithm 1 gives a formal description of Lemke's algorithm applied to LCP (11). Assuming the input parameters $\boldsymbol{U}$, $\boldsymbol{W}$, and $\boldsymbol{L}$ have no polynomial relation among them, Theorem 5 guarantees that any vertex with $z > 0$ is nondegenerate. Therefore, a unique double label, say k, such that $y_k = v_k = 0$ always exists. Algorithm 1 pivots at the double label by relaxing one constraint and traveling along the corresponding edge of $\mathcal{P}$ to the next vertex solution. We prove the following theorem in the next section.

Theorem 7. *If the input parameters* $\boldsymbol{U}$, $\boldsymbol{W}$, *and* $\boldsymbol{L}$ *have no polynomial relation among them, then Algorithm 1 terminates at a competitive equilibrium in finite time.*

Algorithm 1 (Algorithm for Competitive Equilibrium of a Mixed Manna)

```
Data: A, q
Result: A competitive equilibrium
S ← S_0;
while z > 0 do
  | Let k be the double label in solution S, that is, y_k = v_k = 0.
  | if v_k just became 0 then
  |   | Pivot by relaxing y_k = 0.
  | else
  |   | Pivot by relaxing v_k = 0.
  | end
  | Let S′ be the next vertex solution to LCP (11) reached, S ← S′;
end
```

4.1. Convergence of Lemke's Algorithm

We now show that Algorithm 1 always finds an equilibrium when the instance contains at least one good and satisfies the strong connectivity assumption as defined in Section 2.2. Note that the strong connectivity assumption is vacuous when all items are bads. For this case, we provide a separate convergence proof in Appendix C without any assumptions. We note that, unlike earlier works that consider only good manna (Eaves [22], Garg et al. [31], our LCP formulation allows for secondary rays and one nonequilibrium solution. This makes the proof that Lemke's algorithm finds a competitive equilibrium significantly more complex.

Let $\mathcal{P}$ be the polyhedron corresponding to LCP (11). To verify that Algorithm 1 terminates at a competitive equilibrium, we need to examine two potential problems. First, we need to show that the algorithm never finds a secondary ray. Second, we need to show that, starting from the primary ray, Algorithm 1 never reaches the degenerate solution at which $p_j = P, \ \forall j \in M$, and $r_i = R, \ \forall i \in N$.

First, we consider secondary rays. Recall that a ray $\mathcal{R}$ is a unbounded edge of $\mathcal{P}$ incident to the vertex $(\boldsymbol{y}^*, z^*)$ with $z^* > 0$:

$$\mathcal{R} = \{[\boldsymbol{y}^*, z^*] + \alpha[\boldsymbol{y}', z'] \mid \forall \alpha \geq 0\}.$$

Clearly, all points on $\mathcal{R}$ solve LCP (11). Algorithm 1 begins at the primary ray S_0, and all other rays are called secondary. The major issue is that, if Algorithm 1 finds a secondary ray, then it fails to terminate. Observe that setting $p_j = P$ for some $j \in M^-$ leads to secondary rays. Suppose we set $p_j = P$ for some subset of bads $B \subseteq M^-$ and $p_j = 0$ otherwise and make all other variables $(\boldsymbol{f}, \boldsymbol{r}, \boldsymbol{s}) = \boldsymbol{0}$. Then, we may select sufficiently large z^* to satisfy all constraints of the form (11a), (11c), and (11e). Let $\boldsymbol{y}^* = (\boldsymbol{p}, \boldsymbol{f}, \boldsymbol{r}, \boldsymbol{s})$ be this vertex solution and consider the ray $\mathcal{R} = [\boldsymbol{y}^*, z^*] + \alpha[\boldsymbol{0}, 1]$ incident to $(\boldsymbol{y}^*, z^*)$. It is easily verified that $\mathcal{R}$ solves LCP (11) for all $\alpha > 0$ and, therefore, is a secondary ray. We want to show that the path traced by Algorithm 1 never reaches these problematic vertices.

We begin with a simple observation. Notice that setting $p_j = P$ for any good requires that $z = 0$ by (11c) and (11c′). Therefore, Algorithm 1 stops at a vertex at which any $p_j = P$ for any $j \in M^+$. We follow this result with a few useful facts.

Lemma 7. *Let S be any solution to LCP (11) with $p_j < P, \ \forall j \in M$. Pick any agent $i \in N$ and any item (good or bad) $j \in M$ and let $k = |u_{ij}|$ be i's final segment for item j. If $j \in M^-$, then $s_{ijk} = 0$. If $j \in M^+$ and $p_j > 0$, then $s_{ijk} = 0$.*

Proof. Recall that the length of the final segment (i, j, k) is infinite; however, we set $L_{ijk} = 1 + \epsilon$ for some small $\epsilon > 0$ because there is a unit amount of each item. We consider two cases: j is a bad or a good. First, suppose $j \in M^-$, and for a contradiction, assume $s_{ijk} > 0$. By complementarity Condition (11f′), (11f) holds with equality. Then, $f_{ijk} = L_{ijk}(P - p_j) > 0$ because $p_j < P$ at S. Consider Constraint (11b). From these observations, we see that

$$L_{ijk}(P - p_j) = f_{ijk} \leq \sum_{i', k'} f_{i'jk'} \leq P - p_j,$$

a contradiction because $L_{ijk} > 1$, and $p_j < P$.

Now, suppose $j \in M^+$ and $p_j > 0$. For a contradiction, assume $s_{ijk} > 0$. Again (11f′) requires that (11f) holds with equality so that $f_{ijk} = L_{ijk}(P - p_j) > 0$. Because $p_j > 0$, then (11c′) requires that

$$P - p_j = \sum_{i', k'} f_{i'jk'} + \delta_j z \geq f_{ijk} + \delta_j z = L_{ijk}(P - p_j) + \delta_j z,$$

a contradiction because $L_{ijk} > 1$, $\delta_j, z \geq 0$. □

Lemma 8. *At any solution to LCP (11), $p_j \leq P, \ \forall j \in M$ and $r_i \leq R, \ \forall i \in N$. Further, if $r_i = R$ for some $i \in N$, then $p_j = P, \ \forall j \in M^-$.*

Proof. First, suppose $p_j > 0$ for some $j \in M^-$. Complementarity Condition (11b′) requires that (11b) holds with equality. Thus, $\sum_{i,k} f_{ijk} + p_j = P$. Then, $p_j \leq P$ because P and f_{ijk}'s are nonnegative. The case of $j \in M^+$ follows similarly, using complementarity Condition (11c′).

Next, if $r_i > R$ for some $i \in N$, then (11d) is infeasible because of Lemma 7, a contradiction.

For the second claim, we show the contrapositive. Suppose that $p_j < P$ for any $j \in M^-$ and pick any agent $i \in N$. Recall that $0 < D_{ij1} < \cdots < D_{ijk}$, where k is the final segment of i's utility function for j. By Lemma 7, $s_{ijk} = 0$ so that Constraint (11d) for segment (i, j, k) becomes $D_{ijk} r_i - p_j \leq D_{ijk} R - P$. Because $p_j < P$, it follows that $r_i < R$. □

Lemma 9. *Starting from the primary ray, if Algorithm 1 reaches a vertex at which $p_j = P$ for some good $j \in M^+$, then $p_{j'} = P$ for all other items $j' \in M$, $r_i = R$, for all $i \in N$, and $z = 0$.*

Proof. For a contradiction, let T be the vertex solution to LCP (11) in which $p_j = P$ for some good j for the first time and assume that $p_{j'} < P$ for some item $j' \in M$. Let S be the vertex that precedes T starting from the primary ray and E be the edge between S and T. Note that such S exists because we start from the primary ray at which $\boldsymbol{p} = \boldsymbol{r} = \boldsymbol{0}$. At T, $z = 0$ because of (11c), which implies that $z \to 0$ along E.

Let M_1 be the set of goods for which $p_j \to P$ on E and $N_1 = \{i \in N : \exists j \in M_1 \text{ s.t. } U_{ijk} > 0, \ k = |u_{ij}|\}$ be the set of agents that are nonsatiated for some good in M_1.

Claim 1. *At T, $r_i = R, \ \forall i \in N_1$.*

Proof. Let $j \in M_1$ and let $i \in N_1$ be an agent that is not satiated for good j. Let (i, j, k) be i's final segment for good j. Note that $p_j > 0$ on E so that p_j can increase to P. By Lemma 7, $s_{ijk} = 0$. Consider Constraint (11e) for this segment on edge E

$$U_{ijk}(R-r_i)-(P-p_j)-z\le 0. \tag{13}$$

Along E, both $z\to 0$ and $p_j\to P$. Therefore, (13) implies that $r_i\to R$ because $U_{ijk}>0$. □

Claim 2. *If $p_j\to P$ for some good $j\in M^+$, then $p_{j'}\to P,\ \forall j'\in M^-$.*

Proof. Because item j is a good, at least one agent, say i, is nonsatiated for j. Therefore, by Claim 1, $r_i\to R$ on E. Consider any $j'\in M^-$. Let $k'=|u_{ij'}|$ be i's final segment of j'. By Lemma 7, $s_{ij'k'}=0$ on E. Then, Constraint (11d) requires that

$$(P-p_{j'})\le D_{ij'k'}(R-r_i),$$

which implies that $p_{j'}\to P$ because $r_i\to R$. □

Claim 3. *The agents of N_1 purchase no items at T, that is, $f_{ijk}=0,\ \forall j,k,\ \forall i\in N_1$.*

Proof. At T, $p_j=P,\ \forall j\in M^-$ by Claim 2. Therefore, (11b′) requires that $\sum_{i,k}f_{ijk}+p_j=P,\ \forall j\in M^-$ at T. It follows that no agent purchases any bad at T, that is, $f_{ijk}=0\ \forall i,k,\ \forall j\in M^-$. A similar argument shows that no agent purchases any goods $j\in M_1$ at T.

Let $i\in N_1$, and j be any good such that $p_j<P$ at T. For a contradiction, suppose i purchases j at T, that is, at least $f_{ij1}>0$. Then, (11e′) requires that $(P-p_j)+s_{ij1}=0$ because $r_i=R$ and $z=0$ at T. Thus, we obtain a contradiction because $s_{ij1}\ge 0$ and $p_j<P$. Therefore, the agents of N_1 purchase no items (bads or goods) at T. □

Claim 4. *Agents of N_1 are not endowed with any fraction of any good j with $p_j<P$, that is, $\forall i\in N_1$, $W_{ij}=0$ for all $j\in M_0=M^+\backslash M_1$. Therefore, the budget of each agent $i\in N_1$ is equal to zero at T.*

Proof. At T, the following conditions hold for all $i\in N_1$. First, $r_i=R$ by Claim 1. Then, (11a′) requires that (11a) holds with equality. Next, Claim 2 shows that $p_j=P,\ \forall j\in M^-$, and Claim 3 states that $f_{ijk}=0,\ \forall j,k$. Recalling, that $z=0$ at T, then (11a) simplifies to

$$\sum_{j\in M_0}W_{ij}(P-p_j)=0.$$

Clearly, $W_{ij}=0,\ \forall j\in M_0$ for any $i\in N_1$ because $p_j<P,\ \forall j\in M_0$. It follows that agents of N_1 are only endowed with items in $M^-\cup M_1$. All of these items have price $|P-p_j|=0$ at T. Thus, the budget of agents in N_1 equals zero at T. □

We now prove the lemma. Suppose $p_j<P$ for some item at T. Claim 2 shows that $p_j=P,\ \forall j\in M^-$. Therefore, $j\in M_0$. Define $N_0=N\backslash N_1$. Observe that $|N_0|>0$; otherwise, $|M_0|=0$ by Claim 4. It follows from Claim 1 that any agent $i\in N_0$ is satiated for all $j\in M_1$, that is, the final segment (i,j,k) has $U_{ijk}=0$. Further, the agents of N_1 start with only goods of M_1 by Claim 4. Therefore, in the economic graph described in Section 2.2, there are no edges from the agents of N_0 to any agents of N_1. That is, the economic graph is not strongly connected, a contradiction. Therefore, $|M_0|=0$, $N_1=N$, and $p_j=P$ for all j at T. By Claim 1, $r_i=R,\ \forall i\in N$ at T. □

Next, we show that, starting from the primary ray, Algorithm 1 never reaches secondary rays at which $p_j=P,\ \forall j\in S\subset M^-$, whereas $p_j<P,\ \forall j\in M^-\backslash S$. For this, we first prove the following lemma.

Lemma 10. *Starting from the primary ray, if Algorithm 1 reaches a vertex at which $p_j=P$ for some bad $j\in M^-$, then $p_{j'}=P$ for all $j'\in M^-$.*

Proof. For the sake of contradiction, suppose T is the solution to LCP (11) in which $p_j=P$ for some bad $j\in M^-$ for the first time. Now, consider the vertex $S=(\boldsymbol{p},\boldsymbol{f},\boldsymbol{r},\boldsymbol{s},z)$, which precedes T. That is, Algorithm 1 pivots at the vertex S and travels along the edge E to T.

At S, $0\le p_j<P,\ \forall j\in M^-$ because T is the first time $p_j=P$ for some $j\in M^-$. In addition, complementarity Condition (11b′) requires that Constraint (11b) holds with equality for bad j along the entire edge E so that p_j may increase to P. Then, the conditions $\sum_{i,k}f_{ijk}+p_j=P$ and $p_j<P$ imply that at least one agent, say i, spends on some segment (i,j,k) along E. Recall that we select R large enough that the right-hand side of (11d) is positive for all segments (i,j,k). Observe that this implies $r_i>0$; otherwise, (11d) holds with strict inequality, which forces $f_{ijk}=0$, $\forall j,k$ by complementarity Condition (11d′). Thus, segment (i,j,k) is either forced or flexible for i, $r_i>0$, and (11d) holds with equality for segment (i,j,k) along E.

Let j' be a bad such that $p_{j'}<P$ at T. By Lemma 7, i's final segment $k'=|u_{ij'}|$ has $s_{ij'k'}=0$. Because (11d) holds along E for segment (i,j,k), then $D_{ijk}(R-r_i)=P-p_j-s_{ijk}<P-p_j$. Also, because $s_{ij'k'}=0$, then $D_{ij'k'}(R-r_i)\ge P-p_{j'}$

holds along E. Equivalently, we have

$$0 < \frac{D_{ijk}}{(P - p_j)} \le \frac{D_{ijk}}{(P - p_j) - s_{ijk}} = \frac{1}{R - r_i} \le \frac{D_{ij'k'}}{(P - p_{j'})}.$$

Thus, we obtain a contradiction because $p_j \to P$ but $p_{j'} < P$. □

Lemma 10 implies that we cannot have $p_j = P$ for some subset of bads, whereas $p_j < P$ for all remaining bads. We still need to rule out the case in which $p_j = P,\ \forall j \in M^-$. The argument follows similar reasoning to that of Lemma 10. Setting $p_j = P$, that is, the price of all bads equals zero, requires that at least one agent purchases some bad as $p_j \to P$, sending $ppb \uparrow \infty$. The more complicated portion of the proof lies in showing that this agent also must purchase some goods. However, if $p_j < P,\ \forall j \in M^+$, then bpb remains bounded. This gives a contradiction because $bpb \ge ppb$ whenever an agent purchases both bads and goods.

Lemma 11. *Starting from the primary ray, if Algorithm 1 reaches a vertex at which $p_j = P,\ \forall j \in M^-$, then $p_{j'} = P,\ \forall j' \in M^+$ and $z = 0$.*

Proof. For a contradiction, let T be a solution to LCP (11) in which $p_j = P,\ \forall j \in M^-$ for the first time, but $p_j < P,\ \forall j \in M^+$. Note that Lemma 9 shows that $p_j < P,\ \forall j \in M^+$; otherwise, $p_j = P,\ \forall j \in M$. Let S be the vertex that precedes T.

On E, $p_j > 0,\ \forall j \in M^-$ so that p_j may increase to P. Then, the conditions $\sum_{i,k} f_{ijk} + p_j = P$ and $p_j < P,\ \forall j \in M^-$ imply that at least one agent, say i, spends in the agent's first segment $(i,j,1)$ for some bad j. Note that $r_i > 0$; otherwise, (11d) holds with strict inequality, and so (11d′) requires $f_{ijk} = 0$ for all bads. Thus, segment $(i,j,1)$ is either forced or flexible for i, $r_i > 0$, and (11d) holds with equality for segment $(i,j,1)$ along edge. We want to show that these conditions imply that the agent also purchases some good.

Observe that, on the edge E from S to T, every agent's budget eventually becomes strictly positive because $p_j \to P,\ \forall j \in M^-$. Fix $\epsilon > 0$ and pick a point T' on E so that $2|M^-|\max_{j \in M^-}(P - p_j) < \epsilon$. At T', it follows that

$$\sum_{k,\, j \in M^-} |f_{ajk} - W_{aj}(P - p_j)| \le 2|M^-| \max_{j \in M^-}(P - p_j) \le \epsilon,\ \forall a \in N,$$

because $W_{aj} \le 1$, and $\sum_k f_{ajk} \le P - p_j$ by (11b). Recall that $r_i > 0$ so that (11a′) requires that $\sum_{j' \in M^+} W_{ij'}(P - p_{j'}) + z - \sum_{j \in M^-} W_{ij}(P - p_j) + \sum_{k, j \in M^-} f_{ijk} = \sum_{k, j' \in M^+} f_{ij'k}$, or

$$\sum_{j' \in M^+} W_{ij'}(P - p_{j'}) + z - \epsilon \le \sum_{k,\, j' \in M^+} f_{ij'k} \le \sum_{j' \in M^+} W_{ij'}(P - p_{j'}) + z + \epsilon,$$

at T'. Therefore, we must have $f_{ij'k'} > 0$ at least for some segment (i, j', k') of some good j' because $\sum_{j \in M^+} W_{ij}(P - p_j) > 0$, $z \ge 0$, and $\epsilon > 0$ was arbitrary. For this segment, complementarity Condition (11e′) requires that $U_{ij'k'}(R - r_i) = (P - p_{j'}) + z + s_{ij'k'}$. Note that $U_{ij'k'} > 0$ because $(P - p_{j'}) > 0$ on E, and $z, s_{ij'k'} \ge 0$. For the bad j, (11d′) requires $D_{ij1}(R - r_i) = (P - p_j) - s_{ij1}$ because $f_{ij1} > 0$. Further, these conditions hold along the edge from T' to T at which $p_j \to P,\ \forall j \in M^-$. But, then,

$$\frac{U_{ij'k'}}{P - p_{j'} + z + s_{ij'k'}} = \frac{1}{R - r_i} = \frac{D_{ij1}}{(P - p_j) - s_{ij1}} \ge \frac{D_{ij1}}{P - p_j},$$

so that $U_{ij'k'}/(P - p_{j'} + z + s_{ij'k'}) \to \infty$ because $p_j \to P$. Then, we must have $p_{j'} \to P$, $s_{ij'k'} \to 0$, and $z \to 0$, along the edge from T' to T because $s_{ij'k'}, z \ge 0$ and $p_{j'} \le P,\ \forall j \in M$ by Lemma 8. This is a contradiction because $p_{j'} < P$, $\forall j \in M^+$. □

Lemmas 9–11 rule out the possibility of secondary rays for which $p_j = P$ for some $j \in M^-$. We still need to show that Algorithm 1 never reaches the degenerate solution.

Lemma 12. *Starting from the primary ray, Algorithm 1 never reaches the solution $p_j = P,\ \forall j \in M$, and $r_i = R,\ \forall i \in N$ with all other variables, including z, equal to zero.*

Proof. Let T be the degenerate solution. By Lemma 11, Algorithm 1 never reaches a vertex with $p_j = P,\ \forall j \in M^-$, whereas $p_j < P,\ \forall j \in M^+$. Therefore, the only possibility is that *all* p_j are set to P simultaneously.

Consider the vertex S that precedes T. At S, $0 < p_j < P,\ \forall j \in M$ so that p_j's can increase to P. Thus, (11b′) and (11c′) require that (11b) and (11c) hold with equality at S. Summing these equalities over all $j \in M$ shows that the total spending is

$$\sum_{i,k,j\in M^+} f_{ijk} - \sum_{i,k,j\in M^-} f_{ijk} = \sum_{i,j\in M^+} (P - p_j) - \sum_{i,j\in M^-} (P - p_j) - z\sum_{j\in M^+} \delta_j.$$

Note that $z = 0$ at T so that Algorithm 1 stops there. This implies $0 < r_i < R$, $\forall i \in N$, at S so that the r_i can increase to R as required by Lemma 9. Therefore, (11a′) requires that (11a) holds with equality for all $i \in N$. Summing over all i yields

$$\sum_{i,k,j'\in M^+} f_{ij'k} - \sum_{i,k,j\in M^-} f_{ijk} = \sum_{i,j'\in M^+} W_{ij'}(P - p_{j'}) - \sum_{i,j\in M^-} W_{ij}(P - p_j) + zn.$$

Or, because there is a unit amount of each item, that is, $\sum_i W_{ij} = 1$, we see that

$$\sum_{i,k,j'\in M^+} f_{ij'k} - \sum_{i,k,j\in M^-} f_{ijk} = \sum_{j'\in M^+} (P - p_{j'}) - \sum_{j\in M^-} (P - p_j) + zn.$$

This implies that $z(n + \sum_{j\in M^+}\delta_j) = 0$ at S. Thus, $z = 0$ because the $\delta_j > 0$, $\forall j \in M^+$. This means that the algorithm stops at S, which is a competitive equilibrium by Theorem 6. □

Theorem 8. *Starting from the primary ray, Algorithm 1 never reaches a secondary ray.*

Proof. Here, we need to impose conditions on the choices of P and R. After fixing any $P \in \mathbb{R}_+$, select R large enough to ensure that the right-hand side of (11d) is positive, that is, $D_{ijk}R - P > 0$ for all segments (i, j, k), $\forall i,k$, $\forall j \in M^-$, and that the right-hand side of (11e) is negative, that is, $P - U_{ijk}R < 0$ for all segments (i, j, k), $\forall i,k$, $\forall j \in M^+$. Recall that the ray $\mathcal{R} = \{[\boldsymbol{y}^*, z^*] + \alpha[\boldsymbol{y}', z'] \mid \forall \alpha \geq 0\}$ begins at the vertex $(\boldsymbol{y}^*, z^*)$ and travels in the direction $(\boldsymbol{y}', z')$, where $\boldsymbol{y} = (\boldsymbol{p}, \boldsymbol{f}, \boldsymbol{r}, \boldsymbol{s})$.

First, we show that $\boldsymbol{y}' = \boldsymbol{0}$, starting with $\boldsymbol{p}' = \boldsymbol{0}$. Consider Constraints (11b) and (11c) and complementarity Conditions (11b′) and (11c′). For a contradiction, suppose $p'_j > 0$ for some $j \in M$. Then, $p'_j > 0$, $\forall \alpha > 0$, so (11b) or (11c) must hold with equality. Because P is fixed, $f'_{ijk} \geq 0$, $\forall i,j,k$, and $z > 0$, then eventually (11b) or (11c) is violated. Therefore, $\boldsymbol{p}' = \boldsymbol{0}$. Similarly, by (11b), $\boldsymbol{f}' = \boldsymbol{0}$, $\forall j \in M^-$. Note that, if $\boldsymbol{p}' = \boldsymbol{0}$, then the price of each item is constant along E. Recall from Lemma 7 that $s_{ijk} = 0$ for the final segment $k = |u_{ij}|$ of any bad. Therefore, $\boldsymbol{r}' = \boldsymbol{0}$; otherwise, (11d) is eventually violated for the final segment (i, j, k) of any bad $j \in M^-$ for any agent $i \in N$. Also, because $\boldsymbol{f}' = \boldsymbol{0}$ for all bads $j \in M^-$, the spending on bads is constant. If $f'_{ijk} > 0$ for some good $j \in M^+$, then z must increase to ensure Inequality (11a) holds. Further, (11e) must hold for this segment (i, j, k) by complementarity Condition (11e′) because $f'_{ijk} > 0$. However, because r'_i and p'_j are constant and $s_{ijk} \geq 0$, (11e) cannot hold with equality as z increases. This shows that $\boldsymbol{p}'$, $\boldsymbol{f}'$, and $\boldsymbol{r}'$ are constant. Observe that these variables determine $\boldsymbol{s}$ by (11d) and (11e). Therefore, $\boldsymbol{s}' = \boldsymbol{0}$. It follows that $z' > 0$; otherwise, no variables change.

Finally, we show $\boldsymbol{y}^* = \boldsymbol{0}$. Notice that, along the ray $\mathcal{R}$, the money earned and spent by each agent remains constant. However, z increases. Thus, complementarity Condition (11a′) implies that $\boldsymbol{r}^* = \boldsymbol{0}$. It follows that (11d) holds with strict inequality for all bads $j \in M^-$, forcing $\boldsymbol{f}^* = \boldsymbol{0}$ for bads $j \in M^-$ by (11d′). Now, (11b′) requires that $\boldsymbol{p}^* = \boldsymbol{0}$ for all bads $j \in M^-$ because $p_j < P$, $\forall j \in M$ and $\boldsymbol{f}^* = \boldsymbol{0}$ for all bads $j \in M^-$. Because z increases, whereas r_i and p_j remain fixed for all goods $j \in M^+$, complementarity Conditions (11c′) and (11e′) require that both $\boldsymbol{p}^*$ and $\boldsymbol{f}^*$ are equal to $\boldsymbol{0}$ for all goods $j \in M^+$. As a result, $\boldsymbol{s}^* = \boldsymbol{0}$ by (11f′) as (11f) holds with strict inequality $\forall j \in M$ because $p_j < P$, $\forall j \in M$. Therefore, $\boldsymbol{y}^* = \boldsymbol{0}$, and the ray is $\mathcal{R} = [\boldsymbol{0}, z^*] + \alpha[\boldsymbol{0}, 1]$, that is, the primary ray. □

Proof of Theorem 7. Theorem 5 shows that every vertex solution to LCP (11) with $\boldsymbol{p} < P$, $\boldsymbol{r} < R$, and $z > 0$ is nondegenerate as long as there is no polynomial relation among $\boldsymbol{U}$, $\boldsymbol{W}$, and $\boldsymbol{L}$. Lemmas 9–11 show that we never reach a vertex at which $p_j = P$ for any $j \in M$ or $r_i = R$ for any $i \in N$ when the instance contains at least one good. We provide a separate proof for the case when all items are bads in Appendix C. Therefore, there is always a unique double label for Algorithm 1 at which to pivot. Theorem 8 establishes that Algorithm 1 never reaches a secondary ray so that eventually it reaches a solution with $z = 0$, $\boldsymbol{p} < P$, and $\boldsymbol{r} < R$, which is an equilibrium by Theorem 6.

4.2. Results

Theorem 7 directly yields the following results on existence, membership in PPAD, and the rational-valued property.

Theorem 9. *If the fair division instance of a mixed manna under SPLC utilities satisfies strong connectivity, as defined in Section 2.2, then*

i. *There exists a competitive allocation, and Algorithm 1 terminates with one. Furthermore, Algorithm 1 finds a rational-valued solution if all input parameters are rational numbers.*

ii. *The problem of computing a competitive allocation is in* PPAD.

Proof. The first part follows from the last section. The second part follows from the result of Todd [51] on orientability of the path followed by a complementary pivot algorithm and is exactly same as the proof of Garg et al. [31, theorem 6.2]. □

Theorem 10. *If the fair division instance of a mixed manna under SPLC utilities satisfies strong connectivity, as defined in Section 2.2, and the input parameters $\boldsymbol{U}$, $\boldsymbol{W}$, and $\boldsymbol{L}$ have no polynomial relation among them, then there is an odd number of competitive equilibria.*

Proof. Because the parameters $\boldsymbol{U}$, $\boldsymbol{W}$, and $\boldsymbol{L}$ have no polynomial relation among them, Theorem 6 shows that all solutions to LCP (11) with $p_j < P,\ \forall j \in M, r_i < R,\ \forall i \in N$, and $z = 0$ are competitive equilibria. Theorem 7 establishes that Algorithm 1 always terminates at one of these solutions. We now argue that all other equilibria are paired up on paths of the polyhedron corresponding to LCP (11).

Theorem 5 shows that every vertex solution of LCP (11) with $p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$ is nondegenerate. Therefore, a unique double label exists. Lemmas 9–12 show that, starting from a solution with $p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$ and traveling along the edge incident to the double label, we always reach another solution in which $p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$. Thus, a set of paths connect these solutions. Moreover, Theorem 8 shows that these paths never reach a secondary ray. Therefore, if one starts from an equilibrium, then the subsequent path of solutions with $p_j < P,\ \forall j \in M$, and $r_i < R,\ \forall i \in N$ must eventually end at a vertex where $z = 0$, that is, another equilibrium. Then, all other equilibria, besides the one found starting from the primary ray, must be paired. Thus, there is an odd number of equilibria. □

5. Strongly Polynomial Bound for All Bads

Devanur and Kannan [17] offer a strongly polynomial-time algorithm for the exchange model for goods with SPLC utilities when the number of either goods or agents is constant, which Garg and Kannan [26] extend to the more general Arrow–Debreu model with production. The approach uses a *cell decomposition* technique and the fact that n hyperplanes in $\mathbb{R}^d$ form at most $O(n^d)$ nonempty regions or cells. Garg et al. [31] adapt this argument to bound the number of fully labeled vertices in their LCP formulation for the exchange model for goods under SPLC utilities. We follow their analysis and obtain a strongly polynomial bound on the runtime for the case of all bads as well.

The main result of this section is the following theorem.

Theorem 11. *If the fair division instance of a mixed manna under SPLC utilities that contains only bads has either constantly many agents or constantly many bads, then Algorithm 1 runs in strongly polynomial time.*

The idea is as follows. Suppose the number of bads, that is, m, is a constant. We decompose $(\boldsymbol{p}, z)$ space, that is, $\mathbb{R}_+^{m+1}$, into cells by a set of polynomially many hyperplanes such that each cell corresponds to unique setting of forced, flexible, and undesirable partitions. Then, we show that each fully labeled vertex maps into a cell by projection. Further, at most two vertices map to any given cell. Consider LCP (11) from Section 3 with $M^- = M$ (i.e., $M^+ = \emptyset$). That is,

$$\forall i \in N: -\sum_{j\in M} W_{ij} p_j - \sum_{k,j\in M} f_{ijk} - z \le -P \sum_{j\in M} W_{ij} \perp r_i, \tag{14a}$$

$$\forall j \in M: \sum_{i,k} f_{ijk} + p_j \le P \perp p_j, \tag{14b}$$

$$\forall j \in M,\ \forall i,k: D_{ijk} r_i - p_j - s_{ijk} \le D_{ijk} R - P \perp f_{ijk}, \tag{14c}$$

$$\forall (i,j,k): f_{ijk} + L_{ijk} p_j \le L_{ijk} P \perp s_{ijk}. \tag{14d}$$

5.1. Constantly Many Bads

We consider $\mathbb{R}_+^{m+1}$ with coordinates $p_1, \ldots, p_m, z$. For each tuple (i, j, j', k, k'), where $i \in N$, $j \ne j' \in M$, $k \le |u_{ij}|$, and $k' \le |u_{ij'}|$, create a hyperplane $D_{ijk}(P - p_{j'}) - D_{ij'k'}(P - p_j) = 0$. This divides $\mathbb{R}_+^{m+1}$ into cells in which each region has one of the signs $\le$, $=$, or $\ge$. For any agent $i \in N$, the sign of each cell gives a partial order on the pain per buck of the agent's segments. Thus, in any cell, we can sort segments (j, k) of agent i in increasing order of pain per buck and create equivalence classes $B^i_1, \ldots, B^i_l$ with same pain per buck. Let $B^i_{<l} = B^i_1 \cup \cdots \cup B^i_{l-1}$ and define $B^i_{\le l}$ and $B^i_{\ge l}$ similarly.

Next, we show how to represent the flexible partition. We further subdivide each cell by adding the hyperplanes $\sum_{(j,k)\in B^i_{<l}} L_{ijk}(P-p_j) = \sum_j W_{ij}(P-p_j) - z$ for each agent $i \in N$ and each of the agent's partitions B^i_l. For each subcell, let $B^i_{l_i}$ be the rightmost partition such that $\sum_{(j,k)\in B^i_{<l_i}} L_{ijk}(P-p_j) < \sum_j W_{ij}(P-p_j) - z$ for agent i. Then, $B^i_{l_i}$ is the agent's flexible partition. Finally, we add the hyperplanes $p_j = 0,\ \forall j \in M$ and $z = 0$ so that we only consider the cells in which $p_j \ge 0$ and $z \ge 0$. Because every vertex on the path followed by Algorithm 1 satisfies $p_j < P$, we only consider the cells in which $p_j < P$. Observe that any vertex $(\boldsymbol{y}, z)$ traced by our algorithm maps to a cell by projecting it onto $(\boldsymbol{p}, z)$ space.

Lemma 13. *Let $\mathcal{P}$ be the polyhedron corresponding to LCP (14). Then, at most two fully labeled vertices of $\mathcal{P}$ map onto any given cell. Further, if two vertices map to the same cell, then they are adjacent.*

Proof. Each fully labeled vertex and each cell correspond to their own settings of forced, flexible, and undesirable partitions for each agent. Therefore, if a vertex maps to a given cell, then these two setting must match. If a vertex $S = (\boldsymbol{p}, \boldsymbol{f}, \boldsymbol{r}, \boldsymbol{s}, z)$ maps to a certain cell, then the following inequalities are satisfied:

- If $p_j > 0$, then $\sum_{i,k} f_{ijk} = P - p_j$; else $p_j = 0$ at S.
- If $\sum_j W_{ij}(P-p_j) - z \ge 0$, then $\sum_j W_{ij}(P-p_j) - \sum_{j,k} f_{ijk} - z = 0$; else $r_i = 0$ at S.
- If $D_{ij'k'}(P-p_j) - D_{ijk}(P-p_{j'}) \ge 0$ for $(j',k') \in B^i_{l_i}$, then $-D_{ijk}(R-r_i) + (P-p_j) - s_{ijk} = 0$; else $f_{ijk} = 0$ at S.
- If $D_{ij'k'}(P-p_j) - D_{ijk}(P-p_{j'}) > 0$ for $(j',k') \in B^i_{l_i}$, then $f_{ijk} = L_{ijk}(P-p_j)$; else $s_{ijk} = 0$.

In each of these complementarity conditions, one inequality is enforced. Therefore, their intersection forms a line. If this line does not intersect $\mathcal{P}$, then no vertex maps to this cell. If it does, then intersection is either a fully labeled vertex or a fully labeled edge on which the solution is fully labeled along the entire edge. In the former case, only the vertex S maps to the cell. In the latter, only the endpoints of the fully labeled edge map to the cell. Clearly, these vertices are adjacent. □

Notice that the total number of hyperplanes we created is strongly polynomial. Therefore, this creates a strongly polynomial number of cells as well.

5.2. Constantly Many Agents

In this case, we consider the space $\mathbb{R}^n_+$ corresponding to the coordinates of $\boldsymbol{r}$. Then, we create a partitioning of segments corresponding to the bads. Besides this change, the remaining analysis is similar.

Every fully labeled vertex $S = (\boldsymbol{p}, \boldsymbol{f}, \boldsymbol{r}, \boldsymbol{s}, z)$ maps to $\mathbb{R}^n_+$ by taking the projection on $\boldsymbol{r}$. Given a fully labeled vertex, for each bad j sort all of its segments (i, j, k) by increasing order of $D_{ijk}(R-r_i)$ and partition them into equivalence classes $B^j_1, \ldots, B^j_l$. Observe that, at this vertex, bad j gets allocated in order of these partitions. If segment $(i,j,k) \in B^j_l$ is allocated, that is, $f_{ijk} > 0$, then all segments in partitions before B^j_l must also be allocated. We call the last allocated partition the flexible segment, all partitions before it forced partitions, and all partitions after it the undesirable partitions of bad j. Suppose that segment (i, j, k) is in the flexible partition of bad j. Then, $D_{ijk}(R-r_i) = P - p_j$; otherwise, all segments in this partition are either undesirable or all of them are forced for the corresponding agents. Therefore, the flexible partition defines the price of each bad.

Now, we decompose the space $\mathbb{R}^n_+$ into cells in a way that captures the segment configuration of each bad. For each tuple (i, i', j, k, k'), where $i \ne i' \in N$, $j \in M$, $k \le |u_{ij}|$, and $k' \le |u_{i'j}|$, we introduce the hyperplane $D_{ijk}(R-r_i) - D_{i'jk'}(R-r_{i'}) = 0$. In any cell, the signs of these hyperplanes gives a partial order of segments (i, k) for each agent i based on $D_{ijk}(R-r_i)$. Sort segments of each bad j in increasing order of $D_{ijk}(R-r_i)$, and partition them into equality classes $B^j_1, \ldots, B^j_l$.

Next, we capture the flexible partition of each bad. If the bad is fully sold, then simply sum the lengths of segments starting from the first until it becomes one. An undersold bad requires more work. If a bad is undersold, then $p_j = 0$. Thus, segments of its flexible partition satisfy $D_{ijk}(R-r_i) = P$. To capture this, we add the hyperplanes $D_{ijk}(R-r_i) - P = 0$, for all (i, j, k). Observe that the flexible partition of a bad is either the partition when it becomes fully sold or where $D_{ijk}(R-r_i) = P$. This can easily be deduced from the signs of the hyperplanes. Finally, we add the hyperplanes $r_i = 0,\ \forall i$ and consider only those cells for which $0 \le r_i < R,\ \forall i$.

From this discussion, it is clear that the fully labeled vertices that map to a given cell may be worked out similarly to Lemma 13. Further, we obtain one equality for each complementarity condition because each cell captures the complete segment configuration, status of bads, and agents of the instance.

Table 1. Experimental results conducted on random instances.

$N \times M \times$ #Seg	Instances	Minimum iterations	Mean iterations	Maximum iterations
$5 \times 5 \times 5$	1,000	85	137.3	297
$10 \times 5 \times 5$	1,000	107	170.9	395
$10 \times 10 \times 5$	1,000	130	369.1	609
$15 \times 15 \times 5$	50	168	750.3	1,393
$20 \times 20 \times 5$	10	1,127	1,398.2	2,001

Lemma 14. *Let $\mathcal{P}$ be the polyhedron corresponding to LCP (14). Then, at most two fully labeled vertices of $\mathcal{P}$ map onto any given cell. Further, if two vertices map to the same cell, then they are adjacent.*

Clearly, our algorithm follows a systematic path rather than a brute force enumeration of every cell configuration as in Branzei and Sandomirskiy [7] and Garg and McGlaughlin [27]. Theorem 11 follows from this discussion because the number of hyperplanes is strongly polynomial in both cases.

Remark 3. It is not clear how to show a strongly polynomial bound for the case of mixed manna when the number of agents (or items) is a constant. This is due to the additional variable z appearing in (11e) (constraint to force an optimal bundle). This makes the *bpb* condition unusable as a segment configuration at an arbitrary fully labeled vertex.

6. Numerical Experiments

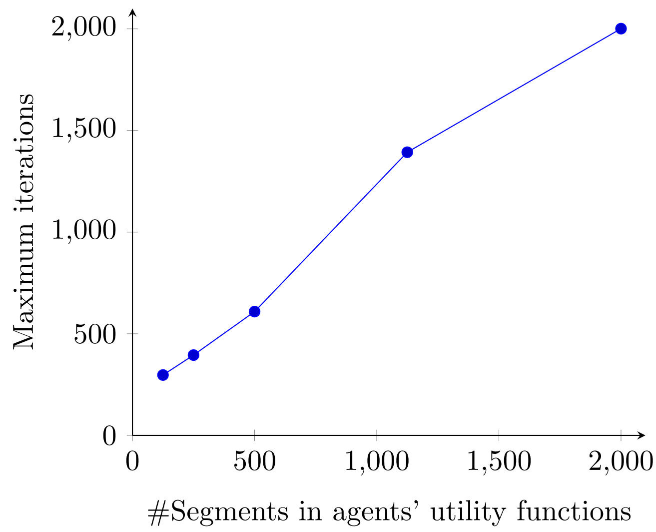
Table 1 summarizes the results of the numerical experiments conducted on randomly generated trials using a Matlab implementation of our algorithm for a bad manna. Note that we used the same number of segments, shown as #Seg in the table, for each agent and each item. We drew the U_{ijk}s, L_{ijk}s, and W_{ij}s uniformly at random from the intervals $[-1, 0]$, $[0, 1/\#\,\text{Seg}]$, and $[0, 1]$, respectively. Then, we rescaled the W_{ij} values to ensure a unit amount of each bad. Finally, for each agent i and each bad j, we sorted the U_{ijk}s in decreasing order to generate SPLC utilities.

Figure 2 compares the maximum number of iterations versus the total number of segments in agents' utility functions, that is, $N \times M \times \#\text{Seg} = \sum_{i,j} |u_{ij}|$. Note that, even in the worst case, the maximum number of iterations is on the order of the total number of segments of the agents' utility functions.

Acknowledgments

The authors are grateful to the anonymous referees for numerous valuable suggestions that have helped to improve the presentation of the paper.

Figure 2. Plot of number of segments in agents' utility functions versus max iterations in Table 1.



Appendix A. Converting Bads into Goods?

Bogomolnaia et al. [3] propose a method to convert a competitive allocation problem with bads into a problem with only goods. Note that their argument only applies to the Fisher setting and uses linear utility functions. The approach relies on the interpretation of leisure as the opposite of work. Therefore, if agent i is assigned an x_{ij} fraction of bad j, then we can equivalently view this as a good representing an exemption from completing a $1-x_{ij}$ fraction of the task.

The reduction from bads to goods proposed by Bogomolnaia et al. [3] is as follows. Assume there are n agents in the competitive division problem. For each bad j, we create $n-1$ units of a good j' representing an exemption from completing bad j. Suppose agent i has utility $D_{ij} < 0$ for bad j; then, i's utility for good j' is an SPLC with two segments. The first segment has slope $|D_{ij}| > 0$ and length $L_{ij} = 1$, and the second segment has slope 0. Note that this means i values up to one unit of exemption to the bad j.

Bogomolnaia et al. [3] state that, given an equilibrium $(\boldsymbol{x}', \boldsymbol{p}')$ in the problem of goods, one can obtain an equilibrium in bads by setting $p^*_j = -p'_j$ and $x^*_{ij} = 1 - x'_{ij}$.

A.1. Counter Example

Consider a competitive division problem with two agents a and b and three bads, 1, 2, and 3. The agents' utility functions are $u_a(x) = -10x_{a1} - 2x_{a2} - x_{a3}$, and $u_b(x) = -x_{a1} - 100x_{a2} - 100x_{a3}$. We create one unit of exemption for each bad. The utility functions for agent a are SPLC in which the first segment has slope (10, 2, 1) for goods 1, 2, and 3, respectively, and are capped at one unit of good. One can verify that prices $\boldsymbol{p}' = (4/3, 1/3, 1/3)$, and allocation $x'_a = (3/4, 0, 0)$ and $x_b = (1/4, 1, 1)'$ are an equilibrium in goods. In bads, this becomes $\boldsymbol{p}^* = (-4/3, -1/3, -1/3)$ with the allocation $x^*_a = (1/4, 1, 1)$ and $x^*_b = (3/4, 0, 0)$. However, this is not a competitive equilibrium because a does not receive the same pbp for all bads. One can check that prices $\boldsymbol{p}^* = (-20/13, -4/13, -2/13)$ along with allocation $x^*_a = (7/20, 4/13, 2/13)$ and $x^*_b = (13/20, 0, 0)$ give an equilibrium.

Appendix B. Approach of Eaves [22] and Garg et al. [31] Gets Stuck on Secondary Rays

Previous works from Eaves [22] and Garg et al. [31] develop complementary pivot algorithms based on Lemke's scheme for good manna under SPLC utilities. The basic structure of our LCP is similar to prior works. However, they use a different change of variables. Both Eaves [22] and Garg et al. [31] use a *lower bound* on prices by making the price of good j $1 + p_j$, where $p_j \geq 0$. Thus, the *minimum* price is one (in absolute value). In addition, Eaves [22] and Garg et al. [31] make no changes to variable $r_i = 1/ppb_i$, where ppb_i is the pain per buck of agent i's flexible segment. In this section, we examine this change of variables when applied to the special case of all bads with linear utilities. The resulting formulation is as follows:

$$\forall i \in N,\ \sum_{j \in M} W_{ij} p_j - \sum_{j \in M} f_{ij} - \epsilon_i z \leq -\sum_{j \in M} W_{ij} \perp r_i, \tag{B.1a}$$

$$\forall j \in M,\ \sum_{i \in N} f_{ij} - p_j \leq 1 \perp p_j, \tag{B.1b}$$

$$\forall i \in N,\ \forall j \in M,\ p_j - D_{ij} r_i - \delta_{ij} z \leq -1 \perp f_{ij}. \tag{B.1c}$$

The constraints have the same interpretation as before: a budget constraint for all agents (B.1a), a constraint on the total spending of agents for each bad (B.1b), and a minimum pain per buck constraint for each agent for each bad (B.1c). Note that we add coefficients ϵ_i and δ_{ij} to z for all terms with a negative right-hand side for two purposes. First, this provides a degree of control over the primary ray, that is, the initial double label, and, therefore, how the algorithm starts. Second, we require δ_{ij}'s coefficients to ensure nondegeneracy of LCP when $z > 0$. To see this, suppose $p_j = 0$ for some $j \in M$, and $r_i = 0$, $\forall i \in N$. Then, by setting $z = 1$, Constraints (B.1c) become tight (hold with equality) for this j, $\forall i \in N$. Thus, there is no unique double label.

We now examine the behavior of Lemke's algorithm when starting from Constraint (B.1a) or (B.1c). We show that, in both cases, the algorithm quickly reaches a secondary ray.

B.1. Starting from (B.1a)

Suppose we select $\epsilon_i = 1$, $\forall i \in N$, and ensure $1/\delta_{ij} < \max_k \sum_j W_{kj}$. By setting $z = \max_k \sum_j W_{kj}$ and all other variables $(\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{f}) = \mathbf{0}$, we obtain a unique double label for Constraint (B.1a) for agent $a = \arg\max_k \sum_j W_{kj}$. Specifically, all Constraints (B.1c) hold with strict inequality.

Lemke's algorithm then fixes $z = \max_k \sum_j W_{kj} = \sum_j W_{aj}$ and increases r_a. Observe that r_a only appears in Constraints (B.1c). However, because $\delta_{aj} z > 1$, and $D_{aj} > 0$, $\forall j \in M$, increasing r_a never makes any Inequality (B.1c) tight for any $j \in M$. That is, we arrived at a secondary ray. Notice that the same problem arises regardless of from which budget Constraint (B.1a) we start (assuming appropriate choice of ϵ_i's and δ_{ij}'s). Therefore, starting from budget Constraint (B.1a) *always* leads to a secondary ray.

B.2. Starting from (B.1c)

Suppose we fix $\epsilon_i = 1,\ \forall i \in N$, and δ_{ij}'s such that $\max_k \sum_j W_{kj} < \max_{i,j} 1/\delta_{ij}$. Then, setting $z = \max_{i,j} 1/\delta_{ij}$ and all other variables $(\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{f}) = \boldsymbol{0}$ yields the unique double label at Constraint (B.1c) for the pair $(a, b) = \arg\min_{i,j} \delta_{ij}$, that is, the agent $a \in N$ and bad $b \in M$ that achieve $\max_{i,j} 1/\delta_{ij}$. Further, all Constraints (B.1a) hold with strict inequality.

Lemke's algorithm fixes $z = 1/\delta_{ab}$ and increases f_{ab} until some other inequality becomes tight. Note that (B.1a) cannot become tight because of our choice of z. Then, (B.1b) becomes tight for bad b. At this point, we may change p_b subject to the following constraints:

$$f_{ab} = 1 + p_b \quad \text{and} \quad 1 + p_b = \delta_{ab} z.$$

We check whether a constraint of form (B.1a) or (B.1c) can become tight.

B.2.1. Starting from Pain per Buck Constraints (B.1c). For any $i \neq a$, we require that (B.1c) becomes tight yet observing the relationship between f_{ab}, p_b, and z. Thus, we need

$$1 + p_b = \delta_{ib} z = \frac{\delta_{ib}}{\delta_{ab}}(1 + p_b) > 1 + p_b,$$

because $\delta_{ab} = \min_{i,j} \delta_{ij}$. Thus, no constraint of the form (B.1c) can become tight.

B.2.2. Budget Constraints (B.1a). For any $i \in N$ (including a), we require that (B.1a) becomes tight yet maintaining the relationship between f_{ab}, p_b, and z. Thus, for $i \neq a$, we need

$$W_{ib} p_b + \sum_j W_{ij} = z = \frac{1 + p_b}{\delta_{ab}},$$

or after rearranging,

$$\underbrace{\Sigma_j W_{ij} - 1/\delta_{ab}}_{<0} = \underbrace{(1/\delta_{ab} - W_{ib})}_{>0} p_b,$$

where the inequalities of the coefficients follow from $1/\delta_{ab} > \max_k \Sigma_j W_{kj} \geq W_{ib}$. However, no value of $p_b \geq 0$ suffices. Similarly, if we want (B.1a) to become tight for agent a, then we need

$$\underbrace{\Sigma_j W_{aj} - 1/\delta_{ab} - 1}_{<0} = \underbrace{(1/\delta_{ab} - W_{ab} + 1)}_{>0} p_b,$$

and again, no value of $p_b > 0$ works.

B.2.3. Conclusion. The examples demonstrate that, for this relationship between f_{ab}, p_b, and z, no constraints can become tight, that is, we have reached a secondary ray.

Appendix C. Convergence of Algorithm 1 with All Bads

In this section, we prove that Algorithm 1 always converges to an equilibrium in the case of all bads, $M^+ = \emptyset$. The proofs are similar in spirit to the mixed manna case, but there are minor differences in some details. We still show that the algorithm never sets a subset price to zero, that is, $p_j = P,\ \forall j \in \tilde{M} \subset M$; rather, all prices are set to zero simultaneously. However, we cannot rely on Lemma 9 to ensure that $r_i = R,\ \forall i \in N$, as used in Lemma 12, which shows that the algorithm stops at an equilibrium before setting $p_j = P,\ \forall j \in M$. This is the only real difference between the proofs.

Recall LCP (14) of Section 5, which gives the augmented LCP formulation for all bads. Let $k = \arg\max_i \sum_{j \in M} W_{ij}$. Then, we get the primary ray by setting $z = \sum_{j \in M} W_{kj}$ and all other variables equal to zero.

We now show that Algorithm 1 never reaches a secondary ray at which $p_j = P$ for some subset of bads and $z > 0$ and that the algorithm never reaches the degenerate solution at which $p_j = P,\ \forall j \in M$ and all other variables equal to zero.

Note that Lemmas 7, 8, and 10 still hold. Therefore, $p_j \leq P,\ \forall j \in M$, $r_i \leq R,\ \forall i \in N$, and if $p_j = P$ for some $j \in M$, then $p_j = P,\ \forall j \in M$. Thus, the algorithm never reaches a secondary ray at which $p_j = P$ for some subset of bads and $z > 0$. It remains to show that the algorithm never reaches the degenerate equilibrium at which $p_j = P,\ \forall j \in M$. The idea is similar to Lemma 12. However, we cannot use Lemma 9 to show that $p_j = P,\ \forall j \in M$ implies $r_i = R,\ \forall i \in N$.

Lemma C.1. *Starting from the primary ray, Algorithm 1 never reaches the degenerate solution at which $p_j = P,\ \forall j \in M$, $r_i = R,\ \forall i \in N$, and all other variables equal to zero.*

Proof. Let T be a vertex at which $p_j = P,\ \forall j \in M$, S be the vertex that precedes T, and E be the edge between S and T. At S, $p_j > 0$ so that all $p_j \to P$ on E. Therefore, complementarity Condition (14b′) requires that (14b) holds with equality on E, $\Sigma_{i,k} f_{ijk} = P - p_j,\ \forall j \in M$. Because $p_j < P$ at S, this requires that, for each bad $j \in M$, at least one agent, say i, purchases some of this bad, that is, $f_{ijk} > 0$. Then, complementarity Condition (14c′) requires that (14c) is tight. Observe that this implies that $r_i > 0$; otherwise, (14c) holds with strict inequality for all segments (i, j, k). Therefore, for this agent, (14a) holds with equality on E by complementarity Condition (14a′).

We argue that $r_i > 0$, $\forall i \in N$. If this condition holds, then (14a) is tight $\forall i \in N$ and (14b) is tight $\forall j \in M$. Summing over all of the constraints yields

$$\sum_j P - p_j = \sum_j W_{ij}(P - p_j) = \sum_{i,j,k} f_{ijk} + nz = \sum_j P - p_j + nz,$$

at S because $\sum_i W_{ij} = 1$. Then, $z = 0$ at S, which is a competitive equilibrium by Theorem 3.

For a contradiction, assume that $r_k > 0$ for some strict subset of agents $k \in N_1 \subset N$. Note that, for all agents $i \in N_0 = N \backslash N_1$, (14c) holds with strict inequality because $r_i = 0$, and therefore, complementarity Condition (14b′) requires that $f_{ijk} = 0$ for all segments (j, k) for all $i \in N_0$. Further, because $p_j > 0$, $\forall j \in M$, at S, then (14b′) requires that (14b) is tight for all $j \in M$. Then, we see that $\sum_{j,k,i \in N_1} f_{ijk} = \sum_{i,j,k} f_{ijk} = \sum_j (P - p_j)$.

Next, observe that (14a) is tight for all $i \in N_1$ by complementarity Condition (14a′). Therefore, $\sum_{j,i \in N_1} W_{ij}(P - p_j) = \sum_{j,k,i \in N_1} f_{ijk} + |N_1|z$. Also, because every agent is endowed with some fraction of at least one bad and $p_j < P$ at S, $\sum_{j,i \in N_1} W_{ij}(P - p_j) < \sum_j (P - p_j)$. Combining this results yields

$$\sum_j (P - p_j) > \sum_{j,i \in N_1} W_{ij}(P - p_j) = \sum_{j,k,i \in N_1} f_{ijk} + |N_1| z = \sum_j (P - p_j) + |N_1| z,$$

at S. Thus, we obtain a contradiction because $p_j < P$, $\forall j \in M$ and $z \geq 0$ at S. □

The only remaining step to show convergence of Algorithm 1 in the case of all bads is to show that the algorithm never reaches a secondary ray at which $\boldsymbol{p} < P$ and $\boldsymbol{r} < R$. However, this follows from the argument of Theorem 8 yet simply ignoring the steps that relate to goods.

Then, Lemmas 10 and C.1 show that, starting from the primary ray, $\boldsymbol{p} < P$ and $\boldsymbol{r} < R$. Specifically, Algorithm 1 never reaches a secondary ray at whih $p_j = P$ for some subset of bads, and it never reaches the degenerate solution. Theorem 8 shows that the algorithm never reaches any other secondary ray. Therefore, eventually, we reach a vertex at which $\boldsymbol{p} < P$, $\boldsymbol{r} < R$, and $z = 0$, which is an equilibrium by Theorem 3.

Endnotes

[1] A similar result is shown for only bad manna (Bogomolnaia et al. [4]). We also refer to an excellent survey article by Moulin [41].

[2] Spliddit (www.spliddit.org) is a user-friendly online platform for computing fair allocation in a variety of problems, which has drawn tens of thousands of visitors in the last five years (Goldman and Procaccia [33]). Spliddit uses linear utilities.

References

[1] Azrieli Y, Shmaya E (2014) Rental harmony with roommates. *J. Econom. Theory* 153:128–137.
[2] Bataille N, Lemaître M, Verfaillie G (1999) Efficiency and fairness when sharing the use of a satellite. *Proc. Fifth Internat. Sympo. Artificial Intelligence Robotics Automation Space*, 465–470.
[3] Bogomolnaia A, Moulin H, Sandomirskiy F, Yanovskaia E (2017) Competitive division of a mixed manna. *Econometrica* 85(6):1847–1871.
[4] Bogomolnaia A, Moulin H, Sandomirskiy F, Yanovskaia E (2019) Dividing bads under additive utilities. *Soc. Choice Welfare* 52(3):395–417.
[5] Brainard W, Scarf H (2000) How to compute equilibrium prices in 1891. Cowles Foundation Discussion Paper 1270, New Haven, CT.
[6] Brams SJ, Taylor AD (1996) *Fair Division—From Cake-Cutting to Dispute Resolution* (Cambridge University Press, Cambridge, UK).
[7] Branzei S, Sandomirskiy F (2019) Algorithms for competitive division of chores. Preprint, submitted July 3, https://arxiv.org/abs/1907.01766.
[8] Budish B, Cantillon E (2010) The multi-unit assignment problem: Theory and evidence from course allocation at Harvard. *Amer. Econom. Rev.* 102(5):2237–2271.
[9] Chaudhury BR, Garg J, McGlaughlin P, Mehta R (2020) Dividing bads is harder than dividing goods: On the complexity of fair and efficient division of chores. Preprint, submitted August 1, https://arxiv.org/abs/2008.00285.
[10] Chaudhury BR, Garg J, McGlaughlin P, Mehta R (2021) Competitive allocation of a mixed manna. Marx D, ed. *Proc. 32nd Sympos. Discrete Algorithms* (SIAM, Philadelphia), 1405–1424.
[11] Chen X, Teng S (2009) Spending is not easier than trading: On the computational equivalence of Fisher and Arrow-Debreu equilibria. Dong Y, Du D-Z, Ibarra OH, eds. *Proc. 20th Internat. Sympos. Algorithms Comput.* (Springer, New York), 647–656.
[12] Chen X, Paparas D, Yannakakis M (2017) The complexity of non-monotone markets. *J. ACM* 64(3):20:1–20:56.
[13] Chen X, Dai D, Du Y, Teng S (2009) Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. *Proc. 50th Sympos. Foundations Comput. Sci.* (IEEE, Atlanta), 273–282.
[14] Chvátal V (1983) *Linear Programming* (W.H. Freeman and Company, New York).
[15] Cottle R, Pang JS, Stone R (1992) *The Linear Complementarity Problem* (Academic Press, Boston).
[16] Dantzig G (1963) *Linear Programming and Extensions* (Princeton University Press, Princeton, NJ).
[17] Devanur N, Kannan R (2008) Market equilibria in polynomial time for fixed number of goods or agents. *Proc. 49th Sympos. Foundations Comput. Sci.* (IEEE, Philadelphia), 45–53.
[18] Devanur N, Garg J, Végh L (2016) A rational convex program for linear Arrow-Debreu markets. *ACM Trans. Econom. Comput.* 5(1):1–13.
[19] Devanur N, Papadimitriou C, Saberi A, Vazirani V (2008) Market equilibrium via a primal–dual algorithm for a convex program. *J. ACM* 55(5):1–18.
[20] Duan R, Mehlhorn K (2015) A combinatorial polynomial algorithm for the linear Arrow-Debreu market. *Inform. Comput.* 243:112–132.

[21] Duan R, Garg J, Mehlhorn K (2016) An improved combinatorial polynomial algorithm for the linear Arrow-Debreu market. Krauthgamer R, ed. *Proc. 27th Sympos. Discrete Algorithms* (SIAM, Philadelphia), 90–106.
[22] Eaves BC (1976) A finite algorithm for the linear exchange model. *J. Math. Econom.* 3(2):197–203.
[23] Eisenberg E (1961) Aggregation of utility functions. *Management Sci.* 7(4):337–350.
[24] Eisenberg E, Gale D (1959) Consensus of subjective probabilities: The Pari-Mutuel method. *Ann. Math. Statist.* 30(1):165–168.
[25] Etkin R, Parekh A, Tse D (2007) Spectrum sharing for unlicensed bands. *Proc. First IEEE Sympos. New Frontiers Dynamic Spectrum Access Networks. IEEE J. Selected Areas Commun.* 25:517–528.
[26] Garg J, Kannan R (2015) Markets with production: A polynomial time algorithm and a reduction to pure exchange. Roughgarden T, Feldman M, Schwarz M, eds. *Proc. 16th Conf. Econom. Comput.* (ACM, New York), 733–749.
[27] Garg J, McGlaughlin P (2020) Computing competitive equilibria with mixed manna. Seghrouchni AEF, Sukthankar G, An B, Yorke–Smith N, eds. *Proc. 19th Conf. Autonomous Agents Multi-Agent Systems* (International Foundation for Autonomous Agents and Multiagent Systems), 420–428.
[28] Garg J, Vazirani VV (2014) On computability of equilibria in markets with production. Chekuri C, ed. *Proc. 25th Sympos. Discrete Algorithms* (SIAM, Philadelphia), 1329–1340.
[29] Garg J, Végh LA (2019) A strongly polynomial algorithm for linear exchange markets. Charikar M, Cohen E, eds. *Proc. 51st Sympos. Theory Comput.* (ACM, New York), 54–65.
[30] Garg J, Mehta R, Vazirani VV (2018) Substitution with satiation: A new class of utility functions and a complementary pivot algorithm. *Math. Oper. Res.* 43(3):996–1024.
[31] Garg J, Mehta R, Sohoni M, Vazirani VV (2015) A complementary pivot algorithm for market equilibrium under separable, piecewise-linear concave utilities. *SIAM J. Comput.* 44(6):1820–1847.
[32] Ghodsi A, Zaharia M, Hindman B, Konwinski A, Shenker S, Stoica I (2011) Dominant resource fairness: Fair allocation of multiple resource types. *Proc. Eighth USENIX Conf. Networked Systems Design Implementation*, 323–336.
[33] Goldman JR, Procaccia AD (2014) Spliddit: Unleashing fair division algorithms. *SIGecom Exchanges* 13(2):41–46.
[34] Hansen KA, Lund TB (2018) Computational complexity of proper equilibrium. Tardos E, Elkind E, Vohra R, eds. *Proc. 19th Conf. Econom. Comput.* (ACM, New York), 113–130.
[35] Jain K (2007) A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM J. Comput.* 37(1):306–318.
[36] Klee V, Minty G (1972) How good is the Simplex algorithm? Shisha O, ed. *Inequalities III* (Academic Press, Boston), 159–175.
[37] Koller D, Megiddo N, von Stengel B (1994) Fast algorithms for finding randomized strategies in game trees. Leighton FT, Goodrich MT, eds. *Proc. 26th Sympos. Theory Comput.* (ACM, New York), 750–759.
[38] Lemke C (1965) Bimatrix equilibrium points and mathematical programming. *Management Sci.* 11(7):681–689.
[39] Lemke C, Howson J (1964) Equilibrium points of bimatrix games. *SIAM J. Appl. Math.* 12(2):413–423.
[40] Moulin H (2003) *Fair Division and Collective Welfare* (MIT Press, Cambridge, MA).
[41] Moulin H (2019) Fair division in the internet age. *Annual Rev. Econom.* 11:407–441.
[42] Orlin J (2010) Improved algorithms for computing Fisher's market clearing prices. Schulman LJ, ed. *Proc. 42nd Sympos. Theory Comput.* (ACM, New York), 291–300.
[43] Pratt JW, Zeckhauser RJ (1990) The fair and efficient division of the Winsor family silver. *Management Sci.* 36(11):1293–1301.
[44] Robertson J, Webb W (1998) *Cake-Cutting Algorithms: Be Fair if You Can* (AK Peters, Natick, MA).
[45] Savani R (2006) Finding Nash equilibria of bimatrix games. Unpublished PhD thesis, London School of Economics and Political Science, London.
[46] Savani R, von Stengel B (2006) Hard-to-solve bimatrix games. *Econometrica* 74(2):397–429.
[47] Sönmez T, Unver U (2010) Course bidding at business schools. *Internat. Econom. Rev.* 51(1):99–123.
[48] Sørensen TB (2012) Computing a proper equilibrium of a bimatrix game. Faltings B, Leyton–Brown K, Ipeirotis P, eds. *Proc. 13th Conf. Econom. Comput.* (ACM, New York), 916–928.
[49] Steinhaus H (1948) The problem of fair division. *Econometrica* 16:101–104.
[50] Su FE (1999) Rental harmony: Sperner's lemma in fair division. *Amer. Math. Monthly* 106(10):930–942.
[51] Todd M (1976) Orientation in complementary pivot algorithms. *Math. Oper. Res.* 1(1):54–66.
[52] Varian H (1974) Equity, envy and efficiency. *J. Econom. Theory* 29(2):217–244.
[53] Vazirani V, Yannakakis M (2011) Market equilibrium under separable, piecewise-linear, concave utilities. *J. ACM* 58(3):1–25.
[54] Végh LA (2014) Concave generalized flows with applications to market equilibria. *Math. Oper. Res.* 39(2):573–596.
[55] Végh LA (2017) A strongly polynomial algorithm for generalized flow maximization. *Math. Oper. Res.* 42(1):179–211.
[56] Vossen TW (2002) Fair allocation concepts in air traffic management. Unpublished PhD thesis, University of Maryland, College Park, MD.
[57] Walras L (1874) *Éléments d'économie politique pure, ou théorie de la richesse sociale (Elements of Pure Economics, or the Theory of Social Wealth)* (Lausanne, Paris), (1899, 4th ed.; 1926, rev ed., 1954, Engl. transl.).
[58] Ye Y (2007) Exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. *Theoretical Comput. Sci.* 378(2):134–142.