



## Methods

# A Strongly Polynomial Algorithm for Linear Exchange Markets

Jugal Garg,<sup>a</sup> László A. Végh<sup>b,\*</sup><sup>a</sup>Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana–Champaign, Urbana, Illinois 61801;<sup>b</sup>Department of Mathematics, London School of Economics, London WC2A 2AE, United Kingdom

\*Corresponding author

Contact: jugal@illinois.edu,  <https://orcid.org/0000-0001-6439-7308> (JG); l.vegh@lse.ac.uk,  <https://orcid.org/0000-0003-1152-200X> (LAV)

Received: September 17, 2019

Revised: August 5, 2021

Accepted: November 25, 2021

Published Online in Articles in Advance:  
February 22, 2022

Area of Review: Optimization

<https://doi.org/10.1287/opre.2021.2258>

Copyright: © 2022 INFORMS

**Abstract.** We present a strongly polynomial algorithm for computing an equilibrium in Arrow-Debreu exchange markets with linear utilities. Our algorithm is based on a variant of the weakly polynomial Duan–Mehlhorn (DM) algorithm. We use the DM algorithm as a subroutine to identify revealed edges—that is, pairs of agents and goods that must correspond to the best bang-per-buck transactions in every equilibrium solution. Every time a new revealed edge is found, we use another subroutine that decides if there is an optimal solution using the current set of revealed edges or, if none exists, finds the solution that approximately minimizes the violation of the demand and supply constraints. This task can be reduced to solving a linear program (LP). Even though we are unable to solve this LP in strongly polynomial time, we show that it can be approximated by a simpler LP with two variables per inequality that is solvable in strongly polynomial time.

**Funding:** Financial support from the Division of Computing and Communication Foundations, National Science Foundation (NSF) [Grants 1755619 and 1942321] and from European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme [Grant ScaleOpt-757481] is gratefully acknowledged.

**Keywords:** market equilibria • linear exchange markets • strongly polynomial algorithm • two-variable-per-inequality systems

## 1. Introduction

Market equilibrium is a central concept in economics to analyze and predict the outcome of agents’ interactions in markets. It has found applications in a variety of domains, even in *nonmarket settings*, which may not involve any exchange of money, but only require the remarkable fairness and efficiency properties of equilibria (see, for example, Varian 1974).

Exchange is one of the most fundamental market models, introduced by Walras (1874). An exchange market is like a barter system, where a set of agents arrive at a market with an initial endowment of divisible goods and have a utility function over allocations of goods. Agents can use their revenue from selling their initial endowment to purchase their preferred bundle of goods. In a market equilibrium, the prices are such that each agent can spend their entire revenue on a bundle of goods that maximizes her utility at the given prices, and all goods are fully sold.

The celebrated result of Arrow and Debreu (1954) shows the existence of an equilibrium for a broad class of utility functions. Computational aspects have already been addressed since the 19th century (see, e.g., Brainard and Scarf 2000), and polynomial time algorithms have been investigated over the last 20 years; see the survey

Codenotti et al. (2004) for early work and the references in Garg et al. (2017) for more recent developments.

In this paper, we study the case where all utility functions are *linear*. Linear market models have been extensively studied since the 1950s; see Devanur et al. (2016) for an overview of earlier work. These models are also appealing from a combinatorial optimization perspective due to their connection to classical network flow models and their rich combinatorial structure. A well-studied special case of the exchange market is the *Fisher market* setting, where every buyer arrives with a fixed budget instead of an endowment of goods. Using network flow techniques, Devanur et al. (2008) gave a polynomial-time combinatorial algorithm that was followed by a series of further such algorithms (for example, Vazirani 2010 and Goel and Vazirani 2011), including strongly polynomial ones (Orlin 2010, Végh 2016). A combinatorial algorithm for the general exchange market was developed later by Duan and Mehlhorn (2015), and no strongly polynomial algorithm has been known thus far.

### 1.1. Strongly Polynomial Algorithms and Rational Convex Programs

Assume that a problem is given by an input of  $N$  rational numbers given in binary description. An algorithm

for such a problem is *strongly polynomial* (see section 1.3 in Grötschel et al. 1988) if it only uses elementary arithmetic operations (additions, multiplications, divisions, and comparisons), and the total number of such operations is bounded by  $\text{poly}(N)$ . Further, the algorithm is required to run in *polynomial space*; that is, the size of the numbers occurring throughout the algorithm remain polynomially bounded in the size of the input. Here, the size of a rational number  $p/q$  with integers  $p$  and  $q$  is defined as  $\lceil \log_2(|p| + 1) \rceil + \lceil \log_2(|q| + 1) \rceil$ .

It is a major open question to find a strongly polynomial algorithm for linear programming. Such algorithms are known for special classes of linear optimization problems. We do not present a comprehensive overview here, but only highlight some examples: systems of linear equations with at most two nonzero entries per inequality (Megiddo 1983, Adler and Cosares 1991, Cohen and Megiddo 1994), minimum cost circulations (for example, Tardos 1985, Goldberg and Tarjan 1989, and Orlin 1993), linear programs (LPs) with bounded entries in the constraint matrix (Tardos 1986, Vavasis and Ye 1996), generalized flow maximization (Olver and Végh 2017, Végh 2017), and variants of Markov Decision Processes (Ye 2005, 2011).

For nonlinear convex optimization, only sporadic results are known. The relevance of certain market equilibrium problems in this context is that they can be described by *rational convex programs*, where a rational optimal solution exists with encoding size bounded in the input size (see Vazirani 2012). This property gives hope for finding strongly polynomial algorithms.

The linear Fisher market equilibrium can be captured by two different convex programs, one by Eisenberg and Gale (1959) and one by Shmyrev (2009). These are special cases of natural convex extensions of classical network flow models (Végh 2013, 2016). In particular, the second model is a network flow problem with a separable convex cost function; Végh (2016) provides a strongly polynomial algorithm for the linear Fisher market using this general perspective.

No such description is known for the exchange-market model. A rational convex program was given in Devanur et al. (2016), but the objective is not separable, and, hence, the result in Végh (2016) cannot be applied. Previous convex programs (Nenakov and Primak 1983, Cornet 1989, Jain 2007) included nonlinear constraints and did not appear amenable for a combinatorial approach (see Devanur et al. 2016 for an overview).

### 1.2. Model

Let  $A$  be the set of  $n$  agents and  $G$  a set of goods. Without loss of generality, we can assume that there is one unit in total of each divisible good and that there is a one-to-one correspondence between agents and goods:

Agent  $i$  brings the entire unit of the  $i$ -th good,  $g_i$ , to the market. If agent  $i$  buys  $x_{ij}$  units of good  $g_j$ , her utility is  $\sum_j u_{ij} x_{ij}$ , where  $u_{ij}$  is the utility of agent  $i$  for a unit amount of good  $g_j$ . Given prices  $p = (p_i)_{i \in A}$ , the bundle that maximizes the utility of agent  $i$  is any choice of *maximum bang-per-buck goods*—that is, goods that maximize the ratio  $u_{ij}/p_j$ . The prices  $p$  and allocations  $(x_{ij})_{i,j \in A}$  form a market equilibrium, if (i)  $\sum_{i \in A} x_{ij} = 1$  for all  $j$ —that is, every good is fully sold; (ii)  $p_i = \sum_{j \in A} p_j x_{ij}$  for all  $i$ —that is, every agent spends her entire revenue; and (iii)  $x_{ij} > 0$  implies that  $u_{ij}/p_j = \max_k u_{ik}/p_k$ —that is, all purchases maximize bang-per-buck.

### 1.3. Algorithms for the Linear Exchange Market

A finite time algorithm based on Lemke (1965) was obtained by Eaves (1976). A necessary and sufficient condition for the existence of equilibrium was described by Gale (1976). The first polynomial-time algorithms for the problem were given by Jain (2007) using the ellipsoid method and by Ye (2008) using an interior point method. The combinatorial algorithm of Duan and Mehlhorn (2015) builds on the algorithm Devanur et al. (2008) for linear Fisher markets. An improved variant was given in Duan et al. (2016) with the current best weakly polynomial running time  $O(n^7 \log^3(nU))$ , assuming all  $u_{ij}$  values are integers between zero and  $U$ . A main measure of progress in these algorithms is the increase in prices. However, the upper bound on the prices depends on the  $u_{ij}$  values; therefore, such an analysis can only provide a weakly polynomial bound. The existence of a strongly polynomial algorithm is described in Duan and Mehlhorn (2015) as a major open question. There are a number of simple algorithms for computing an approximate equilibrium (Devanur and Vazirani 2003, Jain et al. 2003, Garg and Kapoor 2006, Ghyasvand and Orlin 2012), but these do not give rise to polynomial-time exact algorithms.

### 1.4. Overview of the Algorithm

We provide a strongly polynomial algorithm for computing an exact equilibrium in linear exchange markets with running time  $O(n^{10} \log^2 n)$ . Let  $F^*$  denote the set of edges (pairs of agents and goods) that correspond to a best bang-per-buck transaction in every market equilibrium. In the algorithm, we maintain a set  $F \subseteq F^*$  called *revealed edges*, and the main progress is adding a new edge in strongly polynomial time. At a high level, this approach resembles that of Végh (2016), which extends the Orlin (1993) approach for minimum-cost circulations.

In a *money allocation*  $(p, f)$ ,  $(p_i)_{i \in A}$  is a set of prices, and  $(f_{ij})_{i,j \in A}$  represents the amount of money paid for good  $g_j$  by agent  $i$ ;  $f_{ij}$  may only be positive for maximum

bang-per-buck pairs. In the algorithm, we work with a money allocation where goods may not be fully sold and agents may have leftover money; we let  $s(p, f) \in \mathbb{R}^G$  denote surplus vector. Thus,  $\|s(p, f)\|_1$  is the total surplus money left, and  $\|s(p, f)\|_\infty$  is the maximum surplus of any good. We show that if  $f_{ij} > \|s(p, f)\|_1$  in a money allocation, then  $(i, g_j) \in F^*$  (Lemma 2). This is analogous to the method of identifying *abundant arcs* for minimum cost flows by Orlin (1993).

We use a variant of the Duan-Mehlhorn (DM) algorithm to identify revealed edges. We show that the potential  $\phi(p, f) = \|s(p, f)\|_\infty / (\prod_j p_j)^{1/n}$  decreases geometrically in the algorithm; from this fact, it is not difficult to see that an edge with  $f_{ij} > \|s(p, f)\|_1$  appears in a strongly polynomial number of iterations, yielding the first revealed edge. We need to modify the Duan and Mehlhorn (2015) so that, among other reasons, the potential decreases geometrically when we run the algorithm starting with any arbitrary price vector  $p$ ; see Remark 1 for all the differences.

To identify subsequent revealed edges, we need a more flexible framework and a second subroutine. We work with the more general concept of  $F$ -allocations, where the money amount  $f_{ij}$  could be negative if  $(i, g_j) \in F$ . This is a viable relaxation because an  $F$ -equilibrium (namely, a market equilibrium with possibly negative allocations in  $F$ ) can be efficiently turned into a proper market equilibrium, provided that  $F \subseteq F^*$  (Lemma 3). Given a set  $F$  of revealed edges, our *Price Boost* subroutine finds an approximately optimal  $F$ -allocation using only edges in  $F$ . Namely, the subroutine finds an  $F$ -equilibrium if there exists one; otherwise, it finds an  $F$ -allocation that is zero outside  $F$ , and, subject to this, it approximately minimizes  $\phi(p, f)$ . This will provide the initial prices for the next iteration of the DM subroutine. Because DM decreases  $\phi(p, f)$  geometrically, after a strongly polynomial number iterations, it will need to send a substantial amount of flow on an edge outside  $F$ , providing the next revealed edge.

Let us now discuss the *Price Boost* subroutine. The analogous subproblem for Fisher markets in Végh (2016) reduces to a simple variant of the Floyd-Warshall algorithm. For exchange markets, we show that optimizing  $\phi(p, f)$  can be captured by a linear program. A strongly polynomial LP algorithm would therefore immediately provide the desired subroutine. Alas, this LP is not contained in any special class of LP where strongly polynomial algorithms are currently known.

We will exploit the following special structure of the LP. First, we can eliminate the  $f_{ij}$  variables and only work with price variables. We obtain a form where the objective is to maximize the sum of all variables over a feasible set of the form  $P \cap P'$ . The first polyhedron  $P$  is a *monotone two-variable-per-inequality* (M2VPI) system: There is one positive and one nonnegative

variable per inequality. The constraint matrix defining the second polyhedron  $P'$  is what we call a  $Z_+$ -matrix: All off-diagonal elements are nonpositive, but all column sums are nonnegative. This corresponds to a submatrix of the transposed of a weighted Laplacian matrix. A pointwise maximal solution to an M2VPI system can be found in strongly polynomial time using classical results by Megiddo (1983), Adler and Cosares (1991), Cohen and Megiddo (1994), and Hochbaum and Naor (1994). To deal with the constraints defining  $P'$ , we approximate our LP by a second LP that can be solved in strongly polynomial time.

More precisely, we replace the second polyhedron  $P'$  by  $Q$  such that  $\frac{1}{n^2}Q \subseteq P' \subseteq Q$  and that  $Q$  is also a system with one positive and one nonnegative variable per inequality. Thus, an M2VPI algorithm is applicable to maximize the sum of the variables over  $P \cap Q$  in strongly polynomial time. For an optimal solution  $\bar{p}$ , the vector  $\bar{p}/n^2$  is feasible to the original LP, and the objective value is within a factor  $n^2$  of the optimum. For the purposes of identifying a new revealed edge in the algorithm, such an approximation of the optimal  $\phi(p, f)$  value already suffices.

The construction of the approximating polyhedron  $Q$  is obtained via a general method applicable for systems given by  $Z_+$ -matrices. We show that for such systems, Gaussian elimination can be used to generate valid constraints with, at most, two nonzero variables per row. Moreover, we show that the intersection of all relevant such constraints provides a good approximation of the original polyhedron.

The rest of the paper is structured as follows. Section 2 introduces basic definitions and notation. Section 3 describes the overall algorithm by introducing the notion of  $F$ -allocations, the main potential, and the two necessary subroutines. Section 4 proves the lemmas necessary for identifying revealed edges. Section 5 presents the first of these two subroutines, a variant of the Duan-Mehlhorn algorithm. Section 6 shows how the second subroutine, *Price Boost*, can be reduced to solving an LP. Section 7 exhibits the polyhedral approximation result for  $Z_+$ -matrices. Section 8 concludes with some open questions.

A preliminary version of our paper was published as Garg and Végh (2019).

## 2. Preliminaries

For a positive integer  $t$ , we let  $[t] := \{1, 2, \dots, t\}$ , and for  $k < t$ , we let  $[k, t] := \{k, k + 1, \dots, t\}$ . For a vector  $a \in \mathbb{R}^n$ , we let

$$\|a\|_1 := \sum_{i=1}^n |a_i|, \quad \|a\|_2 := \sqrt{\sum_{i=1}^n a_i^2}, \quad \text{and}$$

$$\|a\|_\infty := \max_{i \in [n]} |a_i|,$$



denote the  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$ -norms, respectively. Further, we let  $\text{supp}(a) \subseteq [n]$  denote the support of the vector  $a \in \mathbb{R}^n$ —that is,  $\text{supp}(a) := \{i \in [n] \mid a_i \neq 0\}$ . For a subset  $S \subseteq [n]$ , we let  $a(S) := \sum_{i \in S} a_i$ .

## 2.1. The Linear Exchange Market

We let  $A := [n]$  denote the set of agents,  $G := \{g_1, g_2, \dots, g_n\}$  denote the set of goods, and  $u_{ij} \geq 0$  denote the utility of agent  $i$  for a unit amount of good  $g_j$ . Let  $E \subseteq A \times G$  denote the set of pairs  $(i, g_j)$  such that  $u_{ij} > 0$ ; let  $m := |E|$ . We will assume that for each  $i \in A$ , there exists a  $g_j \in G$  such that  $u_{ij} > 0$ , and for each  $g_j \in G$ , there exists an  $i \in A$  such that  $u_{ij} > 0$ . Hence,  $m \geq n$ .

The goods are divisible, and there is one unit of each in total. Agent  $i$  arrives to the market with her initial endowment comprising exactly the unit of good  $g_i$ . As mentioned in Section 1, the general case with an arbitrary set of goods and arbitrary initial endowments can be easily reduced to this setting; see Jain (2007) and Devanur et al. (2016).

Let  $p = (p_j)_{g_j \in G}$  denote the prices where  $p_j$  is the price of good  $g_j$ . Given prices  $p$ , we let

$$\alpha_i := \max_{g_k \in G} \frac{u_{ik}}{p_k}, \quad \forall i \in A.$$

For each agent  $i \in A$ , the goods satisfying equality here are called *maximum bang-per-buck* (MBB) goods; for such a good  $g_j$ ,  $(i, g_j)$  is called an MBB-edge. We let  $\text{MBB}(p) \subseteq E$  denote the set of MBB-edges at prices  $p$ .

**Definition 1.** Let  $f = (f_{ij})_{i \in A, g_j \in G}$  denote the money flow, where  $f_{ij}$  is the money spent by agent  $i$  on good  $g_j$ . We say that  $(p, f)$  is a *money allocation* if

- i.  $p > 0$ , and  $f \geq 0$ ;
- ii.  $\text{supp}(f) \subseteq \text{MBB}(p)$ ;
- iii.  $\sum_{g_j \in G} f_{ij} \leq p_i$  for every agent  $i \in A$ ;
- iv.  $\sum_{i \in A} f_{ij} \leq p_j$  for every good  $g_j \in G$ .

For the money allocation  $(p, f)$ , the *surplus* of agent  $i$  is defined as

$$c_i(p, f) := p_i - \sum_{g_j \in G} f_{ij},$$

and the surplus of good  $g_j \in G$  is defined as

$$s_j(p, f) := p_j - \sum_{i \in A} f_{ij}.$$

Parts (iii) and (iv) in the definition of the money allocation require that the surplus of all agents and goods are nonnegative. We let  $c(p, f) := (c_i(p, f))_{i \in A}$  and  $s(p, f) := (s_j(p, f))_{g_j \in G}$  denote the surplus vectors of the agents and the goods, respectively. Clearly,  $\|s(p, f)\|_1 = \|c(p, f)\|_1$ , and  $\|s(p, f)\|_\infty \leq \|s(p, f)\|_1 \leq n \|s(p, f)\|_\infty$ .

**Definition 2.** A money allocation  $(p, f)$  is called a *market equilibrium* if  $\|s(p, f)\|_1 = 0$ .

Note that if  $(p, f)$  is a market equilibrium and  $\lambda > 0$  is any positive number, then  $(\lambda p, f)$  is also a market equilibrium. We will refer to this as the *price scaling property of market equilibria*.

## 2.2. Existence of an Equilibrium

A market equilibrium may not exist for certain inputs. For example, consider a market with three agents and three goods, where agent  $i$  brings one unit of good  $g_i$  for  $i = 1, 2, 3$ . Let  $E = \{(1, g_1), (1, g_2), (2, g_3), (3, g_3)\}$ . Clearly, in this market, at any prices, agent 3 will consume the entire  $g_3$ . If  $p_2 > 0$ , then agent 2 will demand a nonzero amount of  $g_3$ , and, therefore, no market equilibrium exists.

A necessary and sufficient condition for the existence of an equilibrium can be given as follows. Let us define the directed graph  $(A, \bar{E})$ , where  $(i, j) \in \bar{E}$  if and only if  $u_{ij} > 0$  (that is, if  $(i, g_j) \in E$ ).

**Lemma 1** (Gale 1976, Devanur et al. 2016). *There exists a market equilibrium if and only if for every strongly connected component  $S \subseteq A$  of the digraph  $(A, \bar{E})$ , if  $|S| = 1$ , then there is a loop incident to the node in  $S$ .*

This condition can be easily checked in strongly polynomial time. Further, it is easy to see that if the above condition holds, then finding an equilibrium in an arbitrary input can be reduced to finding an equilibrium in an input where the digraph  $(A, \bar{E})$  is strongly connected. Thus, we will assume the following throughout the paper:

The graph  $(A, \bar{E})$  is strongly connected. (\*)

## 3. The Overall Algorithm

In this section, we describe the overall algorithm. We formulate the statements that are needed to prove our main theorem:

**Theorem 1.** *There exists a strongly polynomial algorithm that computes a market equilibrium in linear exchange markets in time  $O(n^{10} \log^2 n)$ .*

We start by introducing the concepts of revealed edges,  $F$ -allocations, and balanced  $F$ -flows.

### 3.1. Revealed Edges

Throughout the paper, we let  $F^* \subseteq E$  denote the set of edges  $(i, g_j) \in E$  that are MBB edges for *every* market equilibrium  $(p, f)$ . Throughout the algorithm, we maintain a subset of edges  $F \subseteq F^*$  called the *revealed edge set*. This is initialized as  $F = \emptyset$ .

**Definition 3.** For an edge set  $F \subseteq E$ ,  $(p, f)$  is an  *$F$ -money allocation*, or  *$F$ -allocation* in short, if

- i.  $p > 0$ , and  $f_{ij} \geq 0$  for  $(i, g_j) \in E \setminus F$ ;
- ii.  $\text{supp}(f) \cup F \subseteq \text{MBB}(p)$ ;
- iii.  $\sum_{g_j \in G} f_{ij} \leq p_i$  for every agent  $i \in A$ ;

iv.  $\sum_{i \in A} f_{ij} \leq p_j$  for every good  $g_j \in G$ .

An  $F$ -allocation is called an  $F$ -equilibrium if  $\|s(p, f)\|_1 = 0$ .

Note that  $f_{ij}$  could be negative for  $(i, g_j) \in F$ . A  $\emptyset$ -allocation simply corresponds to a money allocation.

The main progress step in the algorithm is adding new edges to  $F$ . This is enabled by the following lemma; the proof is deferred to Section 4.

**Lemma 2.** *Let  $F \subseteq F^*$ , and let  $(p, f)$  be an  $F$ -allocation. If  $f_{k\ell} > \|s(p, f)\|_1$  for an edge  $(k, g_\ell) \in E$ , then  $(k, g_\ell) \in F^*$ .*

Our algorithm obtains an  $F$ -equilibrium. Whereas an  $F$ -equilibrium is not necessarily a market equilibrium, the following holds true:

**Lemma 3.** *Let  $F \subseteq F^*$ , and assume we are given an  $F$ -equilibrium  $(p, f)$ . Then, a market equilibrium  $(p, f')$  can be obtained in  $O(nm)$  time.*

We let  $\text{FINAL-FLOW}(p)$  denote the algorithm as in the lemma. This is a maximum flow computation in an auxiliary network, as described in the proof in Section 4.

### 3.2. Balanced Flows

Balanced flows play a key role in Duan and Mehlhorn (2015), as well as in previous algorithms for Fisher market models—for example, Devanur et al. (2008), Vazirani (2010), and Goel and Vazirani (2011). We now introduce the natural extension for  $F$ -allocations.

**Definition 4.** Given an edge set  $F \subseteq E$  and prices  $p$ , we say that  $(p, f)$  is a *balanced  $F$ -flow*, if  $(p, f)$  is an  $F$ -allocation that minimizes  $\|s(p, f)\|_1$ , and subject to that, it minimizes  $\|c(p, f)\|_2$ .

**Lemma 4** (Darwish and Mehlhorn 2016, Kamiyama 2019). *Given  $F \subseteq E$  and prices  $p$  such that  $F \subseteq \text{MBB}(p)$ , a balanced  $F$ -flow can be computed in  $O(nm \log(n^2/m))$  time.*

Whereas the original form does not include a revealed set  $F$ , the extension to  $F$ -flows is immediate. We let  $\text{BALANCED}(F, p)$  denote the subroutine guaranteed by the lemma.

### Algorithm 1 (Arrow-Debreu Equilibrium)

**Input:** Set  $A$  of agents, set  $G$  of goods, and utilities  $(u_{ij})_{i \in A, g_j \in G}$   
**Output:** Market equilibrium  $(p, f)$

- 1  $F \leftarrow \emptyset$ ;
- 2 **repeat**
- 3      $(\hat{p}, \hat{f}) \leftarrow \text{BOOST}(F)$  //Theorem 2
- 4      $(p, f) \leftarrow \text{DM}(F, \hat{p})$  //Theorem 3
- 5      $F \leftarrow F \cup \{(i, g_j) \mid f_{ij} > \|s(p, f)\|_1\}$  //Lemma 2
- 6 **until**  $\|s(p, f)\|_1 = 0$
- 7  $f \leftarrow \text{FINAL-FLOW}(p)$  //Lemma 3
- 8 **return**  $(p, f)$

### 3.3. The Algorithm

The overall algorithm is presented in Algorithm 1. The main progress is gradually expanding a revealed edge set  $F \subseteq F^*$ , initialized as  $F = \emptyset$ . Every cycle of the algorithm performs the subroutines  $\text{BOOST}(F)$  and  $\text{DM}(F, \hat{p})$ , and at least one new edge is added to  $F$  at every such cycle. Once an  $F$ -equilibrium is obtained for the current  $F$ , we use the subroutine  $\text{FINAL-FLOW}(p)$  as in Lemma 3 to compute a market equilibrium.

We now introduce the key potential measures used in analysis. For an  $F$ -allocation  $(p, f)$ , we define

$$\phi(p, f) := \frac{\|s(p, f)\|_\infty}{\left(\prod_{j=1}^n p_j\right)^{1/n}}.$$

Note that this is invariant under scaling—that is,  $\phi(p, f) = \phi(\lambda p, \lambda f)$  for any  $\lambda > 0$ . Further,  $(p, f)$  is an  $F$ -equilibrium if and only if  $\phi(p, f) = 0$ . For a given  $F \subseteq F^*$ , we define

$$\Psi(F) := \min\{\phi(p, f) : (p, f) \text{ is an } F\text{-allocation, } \text{supp}(f) \subseteq F\}. \quad (1)$$

**Theorem 2.** *There exists a strongly polynomial time algorithm that for any input  $F \subseteq E$ , returns in time  $O(n^4 \log n)$  an  $F$ -allocation  $(\hat{p}, \hat{f})$  with  $\text{supp}(\hat{f}) \subseteq F$  such that  $\Psi(F) \leq \phi(\hat{p}, \hat{f}) \leq (n-1)^2 \Psi(F)$ .*

The algorithm in the theorem will be denoted as  $\text{BOOST}(F)$  and is described in Section 6. In particular, if  $\Psi(F) = 0$ , then  $\text{BOOST}(F)$  returns an  $F$ -equilibrium.

The second main subroutine  $\text{DM}(F, \hat{p})$  is a variant of the algorithm by Duan and Mehlhorn (2015), described in Section 5. As the input, it uses the prices  $\hat{p}$  obtained in the  $F$ -allocation  $(\hat{p}, \hat{f})$  returned by  $\text{BOOST}(F)$  and outputs an  $F$ -allocation  $(p, f)$  such that either  $\|s(p, f)\|_1 = 0$ —that is, an  $F$ -equilibrium—or it is guaranteed that  $f_{ij} > \|s(p, f)\|_1$  for some  $(i, g_j) \in E \setminus F$  connecting two different connected components of  $F$ . Such an edge  $(i, g_j)$  can be added to  $F$  by Lemma 2.

The existence of the edge will follow from the decrease in the potential  $\phi(p, f)$  during the algorithm  $\text{DM}(F, \hat{p})$ ; this is proved in Section 5.

**Theorem 3.** *There exists a strongly polynomial  $O(n^9 \log^2 n)$  time algorithm that, for a given  $F \subseteq E$  and prices  $\hat{p}$ , computes an  $F$ -allocation  $(p, f)$  such that*

$$\phi(p, f) \leq \frac{\phi(\hat{p}, \tilde{f})}{n^4(m+1)},$$

where  $\tilde{f}$  is the balanced flow computed by  $\text{BALANCED}(F, \hat{p})$ .

The following lemma shows how the decrease in  $\phi(p, f)$  implies the existence of an edge with large flow value connecting two components of  $F$ .

**Lemma 5.** *Let  $(p, f)$  be an  $F$ -allocation with  $\phi(p, f) < \Psi(F)/(n(m+1))$ . Then,  $f_{ij} > \|s(p, f)\|_1$  for at least one edge*

$(i, g_j) \in E \setminus F$  such that  $i$  and  $g_j$  are in two different undirected connected components of  $F$ .

**Proof.** For a contradiction, assume  $f_{ij} \leq \|s(p, f)\|_1$  for every  $(i, g_j) \in E \setminus F$ , where  $i$  and  $g_j$  are in different connected components of  $F$ . Let us modify  $f$  to another flow  $f'$  as follows. We start by setting  $f' = f$ ; then, for every  $(i, g_j) \in E \setminus F$ , where  $i$  and  $g_j$  are in the same component, we reroute  $f_{ij}$  units of flow from  $i$  to  $g_j$  using a path in  $F$ . Such a path may contain both forward and reverse edges; we increase the flow on forward edges and decrease it on reverse edges. Further, we set  $f'_{ij} := 0$  for  $(i, g_j) \in E \setminus F$  if  $i$  and  $g_j$  are in different connected components of  $F$ . The assumption yields

$$\|s(p, f')\|_\infty \leq \|s(p, f')\|_1 \leq (m+1) \|s(p, f)\|_1 \leq n(m+1) \|s(p, f)\|_\infty.$$

Here, we used that setting  $f'_{ij}$  to zero may increase the surplus by  $f_{ij} \leq \|s(p, f)\|_1$ , according to the assumption. Therefore,

$$\phi(p, f') \leq n(m+1)\phi(p, f) < \Psi(F),$$

a contradiction to the definition of  $\Psi(F)$ , because  $\text{supp}(f') \subseteq F$ .  $\square$

Using the above, we are ready to prove Theorem 1.

**Proof of Theorem 1.** By Lemma 5, the number of connected components of  $F$  decreases after every cycle of Algorithm 1; thus, the total number of cycles is  $\leq 2n - 1$ . Consider any cycle. Let  $(\hat{p}, \hat{f})$  denote that  $F$ -allocation returned by  $\text{BOOST}(F)$  with  $\phi(\hat{p}, \hat{f}) \leq (n-1)^2 \Psi(F)$ , and let  $(\hat{p}, \tilde{f})$  denote the balanced  $F$ -flow at prices  $\hat{p}$ . Then,

$$\|s(\hat{p}, \tilde{f})\|_\infty \leq \|s(\hat{p}, \tilde{f})\|_1 \leq \|s(\hat{p}, \hat{f})\|_1 \leq n \|s(\hat{p}, \hat{f})\|_\infty,$$

because  $(\hat{p}, \tilde{f})$  minimizes  $\|s(\hat{p}, \tilde{f})\|_1$  among all  $F$ -allocations. Therefore,  $\phi(\hat{p}, \tilde{f}) < n^3 \Psi(F)$ . Theorem 3 guarantees that  $\text{DM}(F, \hat{p})$  finds an  $F$ -allocation  $(p, f)$  with  $\phi(p, f) < \Psi(F)/(n(m+1))$ . By Lemma 5,  $F$  is extended by at least one new edge in this cycle. The overall running time estimation is dominated by the running time estimation of the calls to  $\text{DM}$ .  $\square$

#### 4. $F$ -allocations and $F$ -equilibria

This section is devoted to the proof of Lemmas 2 and 3, which enable us to add new edges to the revealed set  $F$ , as well as to convert an  $F$ -equilibrium to an equilibrium.

**Proof of Lemma 2.** The claim is immediate if  $(k, g_\ell) \in F$ ; for the rest of the proof, we therefore assume  $(k, g_\ell) \notin F$ . Let  $(p', f')$  be any market equilibrium; we need to show that  $(k, g_\ell) \in \text{MBB}(p')$ . Every edge in  $F$  is an MBB-edge for  $(p, f)$  by the definition of an  $F$ -allocation and also for  $(p', f')$  because of  $F \subseteq F^*$ . By the price scaling property of market equilibria, we can assume

that  $p'_\ell = p_\ell$ . Let  $T \subseteq G$  be the set of goods whose prices at  $p'$  are at least the prices at  $p$ —that is,  $T := \{g_j \in G \mid p'_j \geq p_j\}$ . Clearly,  $g_\ell \in T$ . Let  $S$  be the set of agents who have MBB edges to goods in  $T$  at  $p'$ —that is,

$$S := \{i \in A \mid \exists g_j \in T, (i, g_j) \in \text{MBB}(p')\}.$$

For a contradiction, suppose  $(k, g_\ell) \notin \text{MBB}(p')$ . First, we show that  $k \notin S$ . Indeed, if there existed a good  $g_j \in T$  such that  $(k, g_j) \in \text{MBB}(p')$ , then we would get  $u_{k\ell}/p'_\ell = u_{k\ell}/p'_\ell < u_{ij}/p'_j \leq u_{ij}/p_j$ , a contradiction to  $(k, g_\ell) \in \text{MBB}(p)$ .

Consider the goods in  $T$  at prices  $p$ . Because no good is oversold, we have

$$\sum_{i \in A, g_j \in T} f_{ij} \leq p(T).$$

Also note that  $f_{ij} \geq 0$  whenever  $i \in A \setminus S$  and  $g_j \in T$ . This is because  $f_{ij}$  could be negative only on edges in  $F$ ; however, for every  $(i, g_j) \in F$  such that  $g_j \in T$ , we must have  $i \in S$ , because  $F \subseteq F^* \subseteq \text{MBB}(p')$  by the definition of  $F^*$ . Therefore, the previous inequality implies

$$\sum_{i \in S, g_j \in T} f_{ij} + f_{kl} \leq p(T).$$

Observe that by the choice of  $T$ , there are no edges  $(i, g_j) \in \text{MBB}(p)$  with  $i \in S$  and  $g_j \in G \setminus T$ . Thus, the first term in the left-hand side is the total money they spend. Because their total surplus is at most  $\|s(p, f)\|_1$ , we obtain

$$p(S) \leq p(T) - f_{kl} + \|s(p, f)\|_1.$$

Using  $f_{kl} > \|s(p, f)\|_1$  and breaking  $p(S)$  into two parts, we can rewrite the above as

$$\sum_{j \in S, g_j \notin T} p_j < \sum_{j \notin S, g_j \in T} p_j. \quad (2)$$

Let us now examine the market equilibrium  $(p', f')$ , where  $f' \geq 0$  and  $f'_{ij} > 0$  is allowed only for the MBB edges at  $p'$ . Because every agent spends their budget exactly at equilibrium, we have

$$p'(S) = \sum_{i \in S, g_j \in G} f'_{ij}.$$

Further, because there are no MBB edges at prices  $p'$  from agents outside  $S$  to goods in  $T$ , we obtain

$$p'(S) = \sum_{i \in S, g_j \in G} f'_{ij} \geq \sum_{i \in S, g_j \in T} f'_{ij} = p'(T).$$

Again, we can rewrite the above as

$$\sum_{j \in S, g_j \notin T} p'_j \geq \sum_{j \notin S, g_j \in T} p'_j. \quad (3)$$

Now, because  $p'_j \geq p_j$  for  $g_j \in T$  and  $p'_j < p_j$  for  $g_j \in G \setminus T$ , (2) and (3) give a contradiction.  $\square$

We formulate a simple corollary that will be needed in the proof of Lemma 3.

**Corollary 1.** *If  $F \subseteq F^*$ , then for every  $F$ -equilibrium  $f$ ,  $\text{supp}(f) \subseteq F^*$ .*

**Proof of Lemma 3.** An  $F$ -equilibrium  $(p, f)$  may not be a market equilibrium because  $f$  can have negative values on some edges in  $F$ . If that is the case, we find a flow  $\tilde{f}$  such that  $(p, \tilde{f})$  is a market equilibrium as follows.

Let us construct the network  $N(p)$  on vertex set  $A \cup G \cup \{s, t\}$ , where  $s$  is a source node and  $t$  is a sink node, and the following set of edges:  $(s, i)$  with capacity  $p_i$  for each  $i \in A$ ,  $(g_j, t)$ , with capacity  $p_j$  for each  $g_j \in G$ , and  $(i, g_j)$  with infinite capacity for each  $(i, g_j) \in \text{MBB}(p)$ . Let us use the algorithm by Orlin (2013) to obtain a maximum  $s - t$  flow  $\tilde{f}$  in  $N(p)$  in time  $O(nm)$ .

If  $\tilde{f}$  has value  $\sum_{g_j \in G} p_j$ , then clearly  $(p, \tilde{f})$  is a market equilibrium. We show that this must indeed be the case. Let  $(p', f')$  be a market equilibrium. By the price-scaling property of market equilibria, we can assume that  $\sum_{g_j \in G} p'_j = \sum_{g_j \in G} p_j$ . If  $p' = p$ , then clearly  $(p, \tilde{f})$  is an equilibrium. For the rest of the proof, assume  $p' \neq p$ , let  $\bar{E} := F \cup \text{supp}(f)$ , and let  $E' := \text{MBB}(p')$ . Corollary 1 and the definition of  $F^*$  imply  $\bar{E} \subseteq F^* \subseteq E'$ .

Let  $\alpha := \max_{g_j \in G} \{p'_j/p_j\}$  and  $T := \{j \in G \mid p'_j/p_j = \alpha\}$ . Let  $S$  be the set of agents who have at least one MBB edge to goods in  $T$  when prices are  $p'$ —that is,

$$S = \{i \in A \mid \exists g_j \in T, (i, g_j) \in E'\}.$$

We must have

$$\sum_{i \in S} p'_i \geq \sum_{g_j \in T} p'_j. \quad (4)$$

Now consider the connected components of the bipartite graph  $(A \cup G, \bar{E})$ . Because  $\bar{E} \subseteq E'$ , it follows that  $p_j/p_k = p'_j/p'_k$  whenever  $j$  and  $k$  are in the same connected component. Thus, if  $T \cap C_i \neq \emptyset$  for a connected component, then  $G \cap C_i \subseteq T$ . This implies that  $S \cup T$  is the union of some connected components  $C_1, \dots, C_\ell$  of  $(A \cup G, \bar{E})$ —that is,  $\bigcup_{k=1}^\ell (G \cap C_k) = T$ , and  $\bigcup_{k=1}^\ell (A \cap C_k) = S$ . At the  $F$ -equilibrium  $(p, f)$ , we have

$$\sum_{i \in S} p_i = \sum_{g_j \in T} p_j. \quad (5)$$

Equations (4) and (5), along with the definition of  $\alpha$ , imply that

$$\alpha \sum_{i \in S} p_i \geq \sum_{i \in S} p'_i \geq \sum_{g_j \in T} p'_j = \alpha \sum_{g_j \in T} p_j = \alpha \sum_{i \in S} p_i.$$

Consequently,  $\sum_{i \in S} p'_i = \sum_{g_j \in T} p'_j$  and  $T = \{g_j \in G \mid j \in S\}$ —that is, the set of goods brought by agents in  $S$  is exactly equal to  $T$ . This further implies that  $f'$  for agents in  $S$  and for goods in  $T$  is supported only on the  $\text{MBB}(p)$  edges between them, and, hence,  $\tilde{f}$  must saturate all agents in  $S$  (and equivalently, all goods in  $T$ ) because the set of MBB edges between  $S$  and  $T$  remains same at  $p$  and  $p'$ . Next, we remove the agents in  $S$  and goods in  $T$  and repeat the same analysis on the

remaining set of agents and goods. This proves that  $(p, \tilde{f})$  is a market equilibrium.  $\square$

## 5. The Duan-Mehlhorn Subroutine

In this section, we present a variant of the Duan and Mehlhorn (2015) algorithm as a subroutine  $\text{DM}(F, \hat{p})$  in Algorithm 2. The input is a revealed edge set  $F$  and prices  $\hat{p}$  such that  $F \subseteq \text{MBB}(\hat{p})$ , and the output is either an  $F$ -equilibrium or an  $F$ -allocation  $(p, f)$ , where  $f_{ij} > \|s(p, f)\|_1$  for some  $(i, g_j) \in E \setminus F$  connecting two different components of  $F$ . The modifications compared with the original DM algorithm are listed in Remark 1. We now provide a description where the subroutine terminates once an arc with  $f_{ij} > \|s(p, f)\|_1$  is identified. The variant as required in Theorem 3 can be obtained by simply by removing the termination condition and letting the algorithm run for  $O(n^6 \log^2 n)$  iterations of the outer loop.

### Algorithm 2 ( $\text{DM}(F, \hat{p})$ )

**Input:** Utilities  $(u_{ij})_{i \in A, g_j \in G}$ , an edge set  $F \subseteq E$ , and prices  $\hat{p}$  with  $F \subseteq \text{MBB}(\hat{p})$ .

**Output:** An  $F$ -equilibrium  $(p, f)$  or an  $F$ -allocation  $(p, f)$  such that  $f_{ij} > \|s(p, f)\|_1$  for an  $(i, g_j) \in E \setminus F$ , where  $i$  and  $g_j$  are in different connected components of  $F$ .

```

1  $p \leftarrow \hat{p}; f \leftarrow \text{BALANCED}(F, p);$  //Lemma 4
2 repeat
3   Sort the agents in decreasing order of surplus—
   that is,  $c_1(p, f) \geq c_2(p, f) \geq \dots \geq c_n(p, f)$ 
4   Find the smallest  $\ell$  for which  $c_\ell(p, f)/c_{\ell+1}(p, f) >$ 
    $1 + 1/n$ , and let  $\ell = n$  when there is no such  $\ell$ .
5    $S \leftarrow [\ell]; \Gamma(S) = \{g_j \in G \mid \exists i \in S : f_{ij} \neq 0\}$ 
6    $\gamma \leftarrow 1$ 
7   repeat
8      $x \leftarrow 1$ ; Define  $p_j \leftarrow x p_j, \forall g_j \in \Gamma(S), f_{ij} \leftarrow x f_{ij},$ 
      $\forall i \in S, \forall g_j \in \Gamma(S)$  //  $c_i(p, f)$  and  $s_j(p, f)$  change
     accordingly
9     Increase  $x$  continuously up from one until one
     of the following events occurs;
10    Event 1: A new edge, say  $(a, g_b)$ , becomes
        MBB //  $a \in S, g_b \notin \Gamma(S)$ 
11    Event 2:  $\min_{i \in S} c_i(p, f) = \max\{\max_{i \in S} c_i(p, f), 0\}$ 
        //Balancing
12    Event 3:  $\gamma x = 1 + 1/(56e^2 n^3)$  //Price-rise
13    if Event 1 occurs then
14       $\tilde{c}_i(p, f) \leftarrow c_i(p, f), \forall i \in S \setminus \{a\}$ 
15       $\tilde{c}_a(p, f) \leftarrow c_a(p, f) - p_b$ 
16       $\tilde{c}_i(p, f) \leftarrow c_i(p, f) + f_{ib}, \forall i \notin S$ 
17      if  $\exists i \in A \setminus S$  s.t.  $(i, g_b) \in F$  or  $\min_{i \in S} \tilde{c}_i(p, f) \leq$ 
         $\max\{\max_{i \in S} \tilde{c}_i(p, f), 0\}$  then
18        break //break from the inner loop
19       $f_{ib} \leftarrow 0, \forall i \in A; f_{ab} = p_b; \Gamma(S) \leftarrow \Gamma(S) \cup \{g_b\}; \gamma \leftarrow \gamma x$ 
20    until Event 2 or 3 occurs;
21     $f \leftarrow \text{BALANCED}(F, p)$ 
22  until either  $f_{ij} > \|s(p, f)\|_1$  for an edge  $(i, g_j) \in E \setminus F$  with  $i$ 
    and  $g_j$  in different components of  $F$ , or  $\|s(p, f)\|_1 = 0$ 
23 return  $(p, f)$ 

```



We call one execution of the outer loop a *phase* and one execution of the inner loop an *iteration*. Algorithm 2 first computes a balanced flow  $f$  using the subroutine  $\text{BALANCED}(F, p)$ , as in Lemma 4. Then, the agents are sorted in decreasing order of surplus. Without loss of generality, we assume that  $c_1(p, f) \geq \dots \geq c_n(p, f)$ . Then, we find the smallest  $\ell$  for which the ratio  $c_\ell(p, f)/c_{\ell+1}(p, f)$  is more than  $1 + 1/n$ . If there is no such  $\ell$ , then we let  $\ell := n$ . Let  $S$  be the set of first  $\ell$  agents, and let  $\Gamma(S)$  be the set of goods for which there is a nonzero flow from agents in  $S$ . Because  $f$  is balanced, the agents outside  $S$  have zero flow to goods in  $\Gamma(S)$ —that is,  $f_{ij} = 0, \forall i \notin S, g_j \in \Gamma(S)$ —and the surplus of every good in  $\Gamma(S)$  is zero. The parameter  $\gamma$  measures the cumulative price increment throughout a phase; we set  $\gamma = 1$  before starting the sequence of inner loops.

Next, the algorithm runs the inner loop, where it increases the prices of goods in  $\Gamma(S)$  and the flow between agents in  $S$  and goods in  $\Gamma(S)$  by a multiplicative factor  $x \geq 1$  until one of three events occurs. Observe that except for the MBB edges  $(i, g_j)$ , where  $i \notin S, g_j \in \Gamma(S)$ , all MBB edges remain MBB with this price change, and the surplus of every good in  $\Gamma(S)$  remains zero. When prices of goods in  $\Gamma(S)$  increase, an edge  $(i, g_j)$  from  $i \in S$  and  $g_j \notin \Gamma(S)$  can become MBB. We need to stop when such an event occurs in order to maintain an  $F$ -allocation; this is captured by Event 1. In Event 2, we stop when the surplus of an agent  $i \in S$  becomes equal to either the surplus of an agent  $i' \notin S$  or zero. Let us note that  $c_i(p, f) \geq 0$  is maintained throughout; we use the expression  $\max_{i \in S} c_i(p, f), 0$  to also cover the possible case  $S = [n]$ . In Event 3, we stop when  $\gamma x$  becomes  $1 + 1/(56e^2n^3)$ .

If Event 1 occurs, then we have a new MBB edge  $(a, g_b)$  from  $a \in S$  to  $g_b \notin \Gamma(S)$ . Using this new edge, it is now possible to decrease the surplus of agent  $a$  and increase the surpluses of agents  $i \in S$  by increasing  $f_{ab}$  and decreasing  $f_{ib}$ . We next check if this can lead to making the surplus of an agent  $i \in S$  and  $i' \notin S$  equal. Observe that it is always possible if there exists an edge  $(i', g_b) \in F$ . If yes, then we break from the inner loop; otherwise, we update flow so that agent  $a$  buys the entire good  $g_b$ , add  $g_b$  to  $\Gamma(S)$ , update  $\gamma$  to  $\gamma x$ , and go for another iteration.

**Lemma 6.** *The number of iterations in a phase is, at most,  $n$ .*

**Proof.** Consider the iterations of a phase. At the beginning of every iteration, the size of  $\Gamma(S)$  grows by one, and, hence, there cannot be more than  $n$  iterations in a phase.  $\square$

When we break from the inner loop, we recompute a balanced flow and then check if either  $\|s(p, f)\|_1$  is zero or there is an edge  $(i, g_j) \notin F$  with  $f_{ij} > \|s(p, f)\|_1$

connecting two different components of  $F$ . If yes, then we return the current  $(p, f)$ ; otherwise, we go for another phase. Next, we show that  $(p, f)$  remains an  $F$ -allocation throughout the algorithm, which implies that the algorithm returns an  $F$ -allocation.

**Lemma 7.** *The output  $(p, f)$  of Algorithm 2 is an  $F$ -allocation.*

**Proof.** We only need to show that  $F \subseteq \text{MBB}(p)$  throughout the algorithm. Observe that an MBB edge  $(i, g_j)$  becomes non-MBB only if  $i \notin S$  and  $g_j \in \Gamma(S)$ , where  $S$  and  $\Gamma(S)$  are obtained with respect to a balanced flow  $f$ . If an edge  $(i, g_j) \in F$  is such that  $i \notin S$  and  $g_j \in \Gamma(S)$ , then it contradicts that  $f$  is a balanced flow because the edges in  $F$  are allowed to carry negative flow.  $\square$

The running time analysis of Algorithm 2 is based on the evolution of the norm  $\|c(p, f)\|_2$  and prices  $p$ . If a phase terminates due to Event 3, then we call it *price-rise*; otherwise, *balancing*. The next two lemmas are crucial that eventually imply that the potential function  $\phi(p, f)$  decreases substantially within a strongly polynomial number of phases.

**Lemma 8.** *In Algorithm 2, the price of every good monotonically increases, and the total surplus—that is,  $\|s(p, f)\|_1$ —monotonically decreases.*

**Proof.** Clearly, the price of every good monotonically increases in Algorithm 2. During a price-increase step,  $s_j(p, f) = 0$  is maintained for every  $g_j \in \Gamma(S)$ , and  $s_j(p, f)$  does not change for  $g_j \in G \setminus \Gamma(S)$ . If the allocation changes during Event 1, then  $s_b(p, f)$  decreases to zero, and the other surpluses remain unchanged. When a balanced flow is recomputed at the end of a phase, then  $\|s(p, f)\|_1$  can only decrease.  $\square$

The proof of the next lemma is an adaptation of the proof in Duan et al. (2016) and is given in the appendix.

**Lemma 9.** *Let  $f$  be a balanced flow at the beginning of a phase, and let  $(p', f')$  be the prices and flow at the end of the phase. Then,*

- i.  $\prod_{j=1}^n p'_j \geq \left(1 + \frac{1}{56e^2n^3}\right) \prod_{j=1}^n p_j$  in a price-rise phase, and
- ii.  $\|c(p', f')\|_2 \leq \|c(p, f)\|_2 / \left(1 + \frac{1}{56e^2n^3}\right)$  in a balancing phase.

**Lemma 10.** *The number of arithmetic operations in a phase of Algorithm 2 is  $O(n^3)$ .*

**Proof.** From Lemma 6, the number of iterations in a phase is, at most,  $n$ . In each iteration, we find the minimum  $x$  where one of the events occur, which takes, at most,  $O(n^2)$  arithmetic operations. If Event 1 occurs, then we define another surplus vector  $\tilde{c}(p, f)$ , and, based on this, we decide to exit from the inner loop. This requires additional  $O(n^2)$  arithmetic operations. In total, each iteration takes  $O(n^2)$  arithmetic operations. The steps before the inner loop, like sorting, etc., take  $O(n \log n)$  arithmetic operations. We compute a



balanced flow after exiting from the inner loop, in time  $O(nm \log(n^2/m))$ , according to Lemma 4. Overall, each phase takes  $O(n^3)$  arithmetic operations because  $m \leq n^2$ .  $\square$

In the next lemma, we show that the potential function  $\phi(p, f)$  decreases by a large factor within a strongly polynomial number of phases. This, together with Lemmas 2 and 5, implies that every major cycle terminates in strongly polynomial time.

**Lemma 11.** *The potential function  $\phi(p, f)$  decreases by a factor of at least  $1/n^\gamma$  in  $O((2 + \gamma)^2 n^6 \log^2 n)$  phases of Algorithm 2 for any  $\gamma > 0$ .*

**Proof.** Every phase of Algorithm 2 is either *price-rise* or *balancing*. Using  $\|s(p, f)\|_1/n \leq \|s(p, f)\|_\infty \leq \|s(p, f)\|_1$ , we have the following inequality:

$$\frac{\|s(p, f)\|_1}{n(\prod_i p_i)^{1/n}} \leq \phi(p, f) = \frac{\|s(p, f)\|_\infty}{(\prod_i p_i)^{1/n}} \leq \frac{\|s(p, f)\|_1}{(\prod_i p_i)^{1/n}}. \quad (6)$$

Recall from Lemma 8 that  $\|c(p, f)\|_1$  is monotonically decreasing, and the prices are monotonically increasing throughout. According to Lemma 9, if there are  $C(2 + \gamma)n^3 \log n$  consecutive balancing phases for some constant  $C > 0$ , then  $\|c(p, f)\|_2$  decreases by a factor of at least  $1/n^{2+\gamma}$ . This further implies that the  $\ell_1$  norm—that is,  $\|s(p, f)\|_1 = \|c(p, f)\|_1$ —decreases by a factor of at least  $1/n^{1.5+\gamma}$ .

Consider now a sequence of  $C^2(2 + \gamma)^2 n^6 \log n$  phases. If this contains  $C(2 + \gamma)n^3 \log n$  consecutive balancing phases, then the statement follows as above. Otherwise, there are at least  $C(2 + \gamma)n^3 \log n$  price-rise phases. In that case, the geometric mean of prices—that is,  $(\prod_i p_i)^{1/n}$ —increases by a factor of at least  $n^{2+\gamma}$ . This, together with (6), proves the claim.  $\square$

**Proof of Theorem 3.** According to the above lemma, if we do not terminate Algorithm 2 in the first iteration when an arc  $(i, g_j) \in E \setminus F$  with  $f_{ij} > \|s(p, f)\|_1$  is found, then the potential  $\phi(p, f)$  decreases by a factor  $n^4(m + 1)$  within  $O(n^6 \log^2 n)$  phases.

For a strongly polynomial algorithm, we also need to keep all intermediate numbers polynomial bit length. For this, we can use the Duan and Mehlhorn (2015) technique by restricting the prices and update factor  $x$  to powers of  $(1 + 1/L)$ , where  $L$  has polynomial bit length. This guarantees that all arithmetic is performed on rational numbers of polynomial bit length. As shown in Duan and Mehlhorn (2015), this does not change the number of iterations of the DM subroutine.  $\square$

**Remark 1.** Compared with the original DM algorithm in Duan and Mehlhorn (2015), Algorithm 2 differs in the following:

1. We handle Event 1 (in line 2) differently than the other two events, and this gives rise to two nested loops, unlike Duan and Mehlhorn (2015), where every event is handled similarly, and there is only one loop.

2. The edges in  $F$  are allowed to carry negative flow, unlike Duan and Mehlhorn (2015), where flow is always nonnegative.

3. We initialize prices to  $p$ , unlike Duan and Mehlhorn (2015), where every price is initialized to one. And we stop once a new edge is revealed.

## 6. A Linear Program for $\Psi(F)$

In this section, we first formulate an LP to compute  $\Psi(F)$ . Then, we introduce the class of  $Z_+$ -matrices and formulate a general statement (Theorem 5) that shows how certain LPs with a  $Z_+$  constraint matrix can be approximated by a two-variable-per-inequality system. We use this to prove Theorem 2. The proof of Theorem 5 will be given in Section 7.

Given  $F \subseteq E$ , we consider the bipartite graph  $(A \cup G, F)$ . Let  $C_1, C_2, \dots, C_t$  denote the connected components that have a nonempty intersection with  $G$ . (In particular, we include all isolated vertices in  $G$ , but not those in  $A$ .) Let  $\gamma_i := |C_i \cap G|$ . Let us fix an arbitrary good in each of these components; for simplicity of notation, let us assume that the fixed good in  $C_i$  is  $g_i$ .

If all edges in  $F$  are forced to be MBB edges, then fixing the price  $p_i$  of  $g_i$  uniquely determines the prices of all goods in  $C_i \cap G$ . Indeed, for any buyer  $k \in C_i \cap A$ , and any goods  $g_\ell, g_{\ell'} \in C_i \cap G$  with  $k\ell, k\ell' \in F$ , we have that  $p_\ell/p_{\ell'} = u_{k\ell}/u_{k\ell'}$ . Consequently, for any  $i \in [t]$ , and for any  $g_\ell \in C_i \cap G$ , we can compute the multiplier  $\theta_{i\ell} > 0$  such that  $p_\ell = \theta_{i\ell} p_i$  whenever all edges in  $F$  are MBB. For an agent  $\ell \in A$ , let  $\rho(\ell) \in [t]$  denote the index of the component containing the good  $g_\ell$  of this agent; that is,  $g_\ell \in C_{\rho(\ell)} \cap G$ , and  $p_\ell = \theta_{\rho(\ell)\ell} p_{\rho(\ell)}$ . Let  $\Theta_i := \sum_{g_\ell \in C_i \cap G} \theta_{i\ell}$ ; the total price of the goods in  $C_i$  is  $\Theta_i p_i$ .

### 6.1. Constructing the LP

The variables  $(p_i)_{i \in [t]}$  uniquely determine the price of every good. We compute  $\Psi(F)$  in terms of these variables. To differentiate between this  $t$ -dimensional price vector and the  $n$ -dimensional price vector of all goods, we say that for a price vector  $\bar{p} \in \mathbb{R}^t$ , the vector  $p \in \mathbb{R}^n$  is the *extension* of  $\bar{p}$ , if  $p_\ell = \theta_{\rho(\ell)\ell} \bar{p}_{\rho(\ell)}$  for all  $\ell \in [n]$  (in particular,  $p_\ell = \bar{p}_\ell$  for  $\ell \in [t]$ ). We also say that the  $F$ -allocation  $(p, f)$  is an extension of  $\bar{p}$ , if  $p$  is the extension of  $\bar{p}$ .

**Example 1.** Throughout this section and the next one, we illustrate the argument with the example in Figure 1. There are 10 agents and 15 edges in  $F$ . The edges in  $F$  are depicted by solid edges with the  $u_{ij}$  values indicated; all these are one, except for  $u_{17} = 2$ . The picture does not include the edges in  $E \setminus F$  except for one example: the dashed line for  $(6, g_8)$  with  $u_{68} = 3$ . There are five connected components, containing goods

$\{g_1, g_6\}, \{g_2, g_7, g_8\}, \{g_3, g_9, g_{10}\}, \{g_4\}$ , and  $\{g_5\}$ , with  $p_6 = p_1$ ,  $p_7 = p_8 = 2p_2$ , and  $p_3 = p_9 = p_{10}$ . Thus,  $\Theta_1 = 2, \Theta_2 = 5, \Theta_3 = 3, \Theta_4 = 1, \Theta_5 = 1$ , and  $\gamma_1 = 2, \gamma_2 = 3, \gamma_3 = 3, \gamma_4 = 1, \gamma_5 = 1$ .

We now formulate linear constraints that ensure that a vector  $\bar{p} \in \mathbb{R}^t$  can be extended to an  $F$ -allocation  $(p, f)$  with  $\|s(p, f)\|_\infty \leq 1$ , and  $\text{supp}(f) \subseteq F$ . The first set of constraints will enforce that all edges in  $F$  are MBB, and the second set will guarantee the existence of a desired money flow  $f$  with the surplus bounds.

First, the edges in  $F$  are MBB if and only if  $u_{kj}/p_j \leq u_{kj'}/p_{j'}$  for any  $k \in A$  and any  $g_j, g_{j'} \in G$  such that  $(k, g_j) \in E, (k, g_{j'}) \in F$ . The  $\theta_{i\ell}$  coefficients already capture that equality holds if  $(k, g_i), (k, g_j) \in F$ . For the rest of the pairs, we can express this constraint in terms of the  $\bar{p}$  variables as

$$\begin{aligned} u_{kj}\theta_{\rho(j)j'}\bar{p}_{\rho(j')} - u_{kj'}\theta_{\rho(j)j}\bar{p}_{\rho(j)} &\leq 0 \\ \forall k, j, j' \in A, (k, g_j) \in E \setminus F, (k, g_{j'}) \in F. \end{aligned} \quad (7)$$

We add a second set of constraints for  $\|s(p, f)\|_\infty \leq 1$ . Because  $f$  is supported on  $F$  and is allowed to be negative, this can be guaranteed if and only if for any component  $C_i, i \in [t]$ , the total price of the goods in  $C_i \cap G$  exceeds the total budget of the agents in  $C_i \cap A$  by at most  $\gamma_i = |C_i \cap G|$ . Recall that given the prices  $\bar{p}$  of the fixed goods, the total price of goods in  $C_i \cap G$  is  $\Theta_i \bar{p}_i$ . We obtain the constraints

$$\Theta_i \bar{p}_i - \sum_{k \in C_i \cap A} \theta_{\rho(k)k} \bar{p}_{\rho(k)} \leq \gamma_i \quad \forall i \in [t]. \quad (8)$$

Let us now define the following LP:

$$\begin{aligned} \max \sum_{i=1}^t \bar{p}_i \\ \text{s.t. constraint sets (7) and (8),} \\ \bar{p} \geq 0. \end{aligned} \quad (P_F)$$

Note that  $\bar{p} = 0$  is a feasible solution. Using LP duality, the above program is unbounded if and only if the next LP has a feasible solution  $\bar{p} \neq 0$ .

constraint set (7),

$$\begin{aligned} \Theta_i \bar{p}_i - \sum_{k \in C_i \cap A} \theta_{\rho(k)k} \bar{p}_{\rho(k)} &\leq 0 \quad \forall i \in [t] \\ \bar{p} &\geq 0. \end{aligned} \quad (P_F^0)$$

**Example 2.** Let us show the formulation for the example in Figure 1. The variables are  $\bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4$ , and  $\bar{p}_5$ . From the Constraint Set (7), we only show the example of  $k=6, j=8$ , and  $j'=3$ :

$$3\bar{p}_3 - 2\bar{p}_2 \leq 0.$$

For the components, we have

$$\begin{aligned} 2\bar{p}_1 - \bar{p}_2 - \bar{p}_3 &\leq 2 \\ 5\bar{p}_2 - \bar{p}_1 - \bar{p}_3 &\leq 3 \\ 3\bar{p}_3 - \bar{p}_1 - 2\bar{p}_2 &\leq 3 \\ \bar{p}_4 - 2\bar{p}_2 - \bar{p}_3 - \bar{p}_5 &\leq 1 \\ \bar{p}_5 - \bar{p}_4 &\leq 1. \end{aligned}$$

**Lemma 12.**

i. Any solution  $\bar{p} \in \mathbb{R}^n$  to  $(P_F)$  can be extended to an  $F$ -allocation  $(p, f)$  with  $\|s(p, f)\|_\infty \leq 1$ , and  $\text{supp}(f) \subseteq F$ .

ii. If  $(P_F)$  is bounded, then there exists a pointwise maximal solution  $\bar{p}^* \in \mathbb{R}^t$ ; that is,  $\bar{p} \leq \bar{p}^*$  for any solution  $\bar{p} \in \mathbb{R}^t$  to  $(P_F)$ . Let  $(p^*, f^*)$  denote the extension of these prices to an  $F$ -allocation with  $\|s(p^*, f^*)\|_\infty \leq 1$ , and  $\text{supp}(f^*) \subseteq F$ . Then,  $\Psi(F) = \phi(p^*, f^*)$ .

iii. Under assumption  $(\star)$ , every nonzero solution to  $(P_F^0)$  is strictly positive. Such a solution can be extended to an  $F$ -equilibrium.

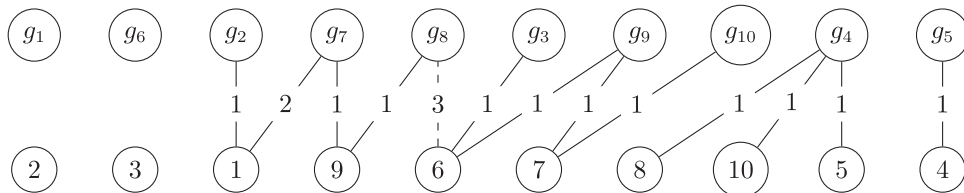
**Proof.** The proof of part (i) was given above. For part (ii), let  $\bar{p}$  and  $\bar{p}'$  be two different solutions, and let  $\bar{p}'' = \max\{\bar{p}, \bar{p}'\}$  for all  $i \in [t]$ . Then, it is easy to see that  $\bar{p}''$  satisfies all Inequalities (7) and (8). (This is true more generally for systems where the transposed of the constraint matrix is pre-Leontief; see Section 6.2.) This implies the existence of a pointwise maximal  $\bar{p}^*$  in case  $(P_F)$  is bounded.

Let  $(p, f)$  be the optimal  $F$ -allocation such that  $\Psi(F) = \phi(p, f)$ . Because  $\phi(p, f) = \phi(\alpha p, \alpha f)$  for any  $\alpha > 0$ , we may assume that  $\|s(p, f)\|_\infty = 1$ . Then,  $(p_i)_{i \in [t]}$  is feasible to  $(P_F)$ , and, therefore,  $p_i \leq p_i^*$  for all  $i \in [t]$ . Consequently,  $\phi(p^*, f^*) \leq \phi(p, f)$ . By definition of  $\Psi(F)$ , we have  $\phi(p^*, f^*) \geq \Psi(F)$ . Thus, equality must hold, which in particular implies  $p_i^* = p_i$  for all  $i \in [t]$ .

Let us now turn to part (iii), when  $(P_F^0)$  has a nonzero solution  $\bar{p} \in \mathbb{R}^t$  (and thus  $(P_F)$  is unbounded). Let  $p \in \mathbb{R}^n$  be the extension of  $\bar{p}$ .

Summing up the second set of constraints in  $(P_F^0)$ , the coefficient of every  $\bar{p}_i$  is nonnegative. Consequently, all

**Figure 1.** Example Problem Setting



these inequalities must hold at equality, implying that in every component  $C_i$ , the total budget of the agents in  $C_i \cap A$  equals the total price of the goods in  $C_i \cap G$ . Further, if  $i \in A$  has no incident edges in  $F$ , then  $p_i = 0$  must hold. If  $p > 0$ , then the union of the components  $C_i$  equals  $A \cup G$ . Then, we can set  $f$  with  $\text{supp}(f) \subseteq F$  such that  $\|s(p, f)\|_1 = 0$ , and such a  $(p, f)$  gives an  $F$ -equilibrium.

It is left to show that  $\bar{p} > 0$ , or, equivalently,  $p > 0$ . Let  $A_0 := \{k \in A : p_k = 0\}$  and assume  $A_0 \neq \emptyset$ . Note that  $A \setminus A_0 \neq \emptyset$  because  $p \neq 0$ . By assumption  $(\star)$ , there exist  $k \in A \setminus A_0$  and  $\ell \in A_0$  such that  $u_{k\ell} > 0$ . There exists at least one edge  $(k, g_j) \in F$ , as otherwise  $p_k = 0$ , as shown above. By (7),  $j \in A_0$  must hold. Let  $C_i$  be the component containing agent  $k$  and good  $g_j$ . Clearly, for every  $g_{j'} \in C_i \cap G$ , we have  $j' \in A_0$ . Hence, the budget of every agent in  $C_i \cap A$  must also equal zero, in particular,  $p_k = 0$ , a contradiction.  $\square$

## 6.2. Monotone Two-Variable-per-Inequality Systems

Let  $\mathcal{M}_2(m, n)$  denote the set of  $m \times n$  rational matrices with at most one positive and at most one negative entry per row. For a matrix  $A \in \mathcal{M}_2(m, n)$ , and an arbitrary vector  $b \in \mathbb{Q}^m$ , the LP  $Ax \leq b$  is called a *monotone two-variable-per-inequality system*. In every such system, whenever the objective function  $\max \sum_i x_i$  is bounded, there exists a *pointwise maximal feasible solution*—that is, a feasible  $x^*$  such that for every feasible solution  $x$ ,  $x \leq x^*$ .

This property holds more generally. Namely, a matrix is called *pre-Leontief* if every column contains at most one positive element. If  $A^\top$  is pre-Leontief, then the system  $A^\top x \leq c$  has a pointwise maximal feasible solution whenever  $\max \sum_i x_i$  is bounded; see Cottle and Veinott (1972). Whereas M2VPI systems are strongly polynomially solvable, as stated in the next theorem, no such algorithm is currently known for the general pre-Leontief setting.

**Theorem 4** (Megiddo 1983, Hochbaum and Naor 1994). *Consider an M2VPI system  $Ax \leq b$  with  $A \in \mathcal{M}_2(m, n)$ . Then, there exists a strongly polynomial  $O(mn^2 \log m)$  time algorithm that finds a feasible solution or concludes infeasibility. Further, if there exists a pointwise maximal feasible solution, the algorithm also finds that one.*

Note that this theorem is not directly applicable to  $(P_F)$ . Whereas the Constraints (7) are of the required form, the Constraints (8) have only one positive coefficient, but possibly multiple negative ones. In what follows, we show that finding an approximate solution to  $(P_F)$  can be reduced to an M2VPI system.

## 6.3. $Z_+$ -Matrices

Let  $M \in \mathbb{R}^{k \times t}$  be the matrix representing the left-hand side of the constraints in (8). That is, for all  $i, j \in [t]$ ,

we let

$$M_{ij} := \begin{cases} \Theta_i - \sum_{k \in C_i \cap A: p(k)=i} \theta_{ik}, & \text{if } i = j, \\ - \sum_{k \in C_i \cap A: p(k)=j} \theta_{jk}, & \text{if } i \neq j. \end{cases} \quad (9)$$

Thus, (8) can be written as  $M\bar{p} \leq \gamma$ , where  $\gamma^\top = (\gamma_1, \dots, \gamma_t)$ .

**Definition 5.** A matrix  $M \in \mathbb{Q}^{k \times t}$  is a  $Z_+$ -matrix if all off-diagonal entries are nonpositive and all column sums are nonnegative. For a nonsquare matrix, by diagonal entries, we mean all entries  $z_{ii}$  for  $1 \leq i \leq \min\{k, t\}$ . We let  $\mathcal{Z}_+(k, t)$  denote the set of  $k \times t$   $Z_+$ -matrices.

Clearly, the matrix  $M$  defined by (9) is in  $\mathcal{Z}_+(t, t)$ . Recall that a matrix is called a  $Z$ -matrix if all off-diagonal entries are nonpositive; the notation reflects the additional requirement on the columns. Further, note that a matrix is a  $Z_+(t, t)$ -matrix if and only if it is the transposed of a weighted Laplacian of a directed graph on  $t$  vertices or if it can be obtained by deleting a row and a column of the transposed of a weighted Laplacian of a directed graph on  $t + 1$  vertices. We will prove the following theorem on LPs with  $Z_+$ -matrices as constraint matrices.

**Theorem 5.** *Given a matrix  $M \in \mathcal{Z}_+(k, t)$  with  $\ell$  nonzero entries, and  $b \in \mathbb{Q}^k$ , with  $b > 0$ , we let*

$$P_M = \{x \in \mathbb{R}^t : Mx \leq b, x \geq 0\}.$$

*Then, in time  $O(\ell(\min\{k, t\})^3)$ , we can construct a matrix  $\bar{M} \in \mathcal{M}_2(\ell', t)$  and  $\bar{b} \in \mathbb{Q}^{\ell'}$  for  $\ell' \leq \ell$  such that*

$$P_M \subseteq \{x \in \mathbb{R}^t : \bar{M}x \leq \bar{b}, x \geq 0\} \subseteq B^2 P_M,$$

*where  $B = \sum_{j=1}^k b_j / \min_{i \in [k]} b_i$ . Further, the size of the entries in  $\bar{M}$  and  $\bar{b}$  are polynomially bounded in the encoding size of the input.*

Here, we use the notation  $\alpha P = \{\alpha x : x \in P\}$  for a set  $P$  and a constant  $\alpha > 0$ . The proof of Theorem 5 will be given in Section 7; we now use it to derive Theorem 2.

**Proof of Theorem 2.** Lemma 12 establishes that computing  $\Psi(F)$  is equivalent to solving the LP  $(P_F)$ . We construct a second LP  $Q_F$  as follows. For the Constraint Set (8) in the form  $M\bar{p} \leq \gamma$ , we apply Theorem 5 to obtain  $\bar{M}\bar{p} \leq \bar{\gamma}$ . Note that  $B \leq n$ , because  $\sum_{i=1}^t \gamma_i = n$ , and  $\gamma_i \geq 1$  for  $i \in [t]$ . Then, we let

$$Q_F := \{\bar{p} \in \mathbb{R}^t : \bar{p} \text{ satisfies (7) and } \bar{M}\bar{p} \leq \bar{\gamma}\}.$$

Let  $P_F$  denote the feasible region of  $(P_F)$ . Using that all right-hand sides in (7) are zero, we see that

$$P_F \subseteq Q_F \subseteq n^2 P_F.$$

Because  $Q_F$  is an M2VPI system, Theorem 4 provides a strongly polynomial algorithm to obtain the prices  $\bar{p}$  maximizing  $\sum_{i=1}^t \bar{p}_i$  over  $Q_F$  or concludes that



this objective is unbounded on  $Q_F$ . In case a finite optimum exists,  $\bar{p}/n^2$  is feasible to  $(P_F)$  and is within a factor  $n^2$  from an optimal solution.

If the objective is unbounded on  $Q_F$ , then we claim that we can get a nonzero solution to  $(P_F^0)$ . Using LP duality, the objective is unbounded on  $Q_F$  if and only if there is a feasible solution  $\bar{p} \neq 0$  to

$$Q_F^0 = \{\bar{p} \in \mathbb{R}^t : \bar{p} \text{ satisfies (7) and } \bar{M}\bar{p} \leq 0\}.$$

Again, Theorem 4 is applicable to find a nonzero solution  $q$ . Suppose  $q \neq 0$  is a solution to  $Q_F^0$ . This implies that  $\alpha q$  is a feasible solution to  $Q_F$  for all  $\alpha \geq 0$ . Because for every feasible solution  $\bar{p}$  to  $Q_F$ ,  $\bar{p}/n^2$  is a feasible solution to  $(P_F)$ , this further implies that  $\alpha q$  is also a feasible solution to  $(P_F)$  for all  $\alpha \geq 0$ . Therefore,  $q$  must be a solution to  $(P_F^0)$ .

The number of nonzero entries in  $M$  is  $\leq 2n$ . Thus, constructing  $\bar{M}$  and  $\bar{\gamma}$  takes  $O(n^4)$  time. We obtain an M2VPI system with  $\leq m + 2n$  constraints and  $\leq n$  variables, and  $m = O(n^2)$ , thus the running time for solving the M2VPI system is  $O(n^4 \log n)$  that dominates the total running time.

Finally, for a strongly polynomial algorithm, we also have to provide polynomial bounds on the encoding lengths of the numbers during the algorithm. The entries of  $M$  are simple expressions of the input parameters  $u_{ij}$ . Then, Theorem 5 guarantees that  $\bar{M}$  and vector  $\bar{\gamma}$  also have bounded encoding length. Thus, the strongly polynomial M2VPI algorithm takes a polynomial size input, and, therefore, the overall algorithm will be strongly polynomial.  $\square$

## 7. Approximating Systems with $Z_+$ -Matrices

This section is dedicated to the proof of Theorem 5. For consistency with the market terminology, we use  $\bar{p} \in \mathbb{R}^t$  as the variables. Recall that we need to show that given a system  $P_M = \{\bar{p} \in \mathbb{R}^t : M\bar{p} \leq \gamma, \bar{p} \geq 0\}$  with  $M \in \mathcal{Z}_+(k, t)$  with  $\ell$  nonzero entries and  $\gamma \in \mathbb{Q}^k, \gamma > 0$ , we can construct a matrix  $\bar{M} \in \mathcal{M}_2(\ell', t)$  and  $\bar{\gamma} \in \mathbb{Q}^{\ell'}$  for  $\ell' \leq \ell$  in  $O(\ell t^3)$  time such that  $P_M \subseteq \{\bar{p} \in \mathbb{R}^t : \bar{M}\bar{p} \leq \bar{\gamma}, \bar{p} \geq 0\} \subseteq B^2 P_M$ , where  $B = \sum_{j=1}^k \gamma_j / \min_{i \in [k]} \gamma_i$ .

Let  $M_i \in \mathbb{R}^t$  denote the  $i$ -th row of the matrix  $M$  for  $i \in [t]$ . We will assume that  $k = t$ —that is,  $M$  is a square matrix. Indeed, if  $k > t$ , then the last  $k - t$  rows only contain nonpositive coefficients. Therefore, for  $i > t$ ,  $M_i \bar{p} \leq 0$  holds for every  $\bar{p} \geq 0$ . If  $t > k$ , then by the  $Z_+$ -property, all entries of the last  $t - k$  columns must be zero, and, thus, we can delete these columns. We further assume that all diagonal entries are strictly positive; if  $M_{ii} = 0$ , then we can remove the  $i$ -th row and  $i$ -th column similarly. Let us also note that every matrix in  $\mathcal{Z}_+(t, t)$  can be obtained in the Form (9), corresponding to a market problem with  $t$  components.

### 7.1. Lower Bounding the Vector $M\bar{p}$

For  $i \in [t]$ , we let  $\lambda_i := \sum_{j \neq i} \gamma_j$  and we let  $\lambda := (\lambda_1, \dots, \lambda_t)^\top$ .

**Lemma 13.** *Let  $M \in \mathcal{Z}_+(t, t)$ . Assume that  $\bar{p} \in \mathbb{R}_+^t$  satisfies  $M\bar{p} \leq \gamma$ . Then, we also have*

$$M\bar{p} \geq -\lambda \geq -(B-1)\gamma.$$

**Proof.** The sum of the rows is  $\sum_{j \in [t]} M_j \geq 0$  because  $M$  is a  $Z_+$ -matrix. Therefore, for any  $i \in [t]$ , we have

$$M_i \bar{p} = \left( \sum_{j \in [t]} M_j \right) \bar{p} - \left( \sum_{j \in [t], j \neq i} M_j \right) \bar{p} \geq - \sum_{j \in [t], j \neq i} \gamma_j = -\lambda_i.$$

The inequality  $\lambda \leq (B-1)\gamma$  follows by the definition of  $B$ .  $\square$

### 7.2. Gaussian Elimination for $Z_+$ -Matrices

We will use Gaussian elimination to generate new constraints. For this purpose, we show that Gaussian elimination on  $Z_+$ -matrices will only add nonnegative multiples of rows to other rows.

**Lemma 14.** *Let  $T \in \mathcal{Z}_+(\ell, t)$ . Then, using Gaussian elimination, we can obtain a matrix  $T' = YT$ , where  $T' \in \mathbb{R}^{\ell \times t}$  is an upper triangular matrix with diagonal entries zero or one and all off-diagonal entries are nonpositive; further, all entries of  $Y \in \mathbb{R}^{\ell \times \ell}$  are nonnegative. If  $T_{ik} < 0$  for some  $k \in [t]$ ,  $i \in [k+1, \ell]$ , then  $T'_{kk} = 1$ .*

**Proof.** Let  $T^{(k)} = Y^{(k)}T$  be the matrix after  $k$  steps in the Gaussian elimination with  $T^{(0)} = T$  and  $Y^{(0)} = I_\ell$ . By induction on  $k$ , we simultaneously show the following:

- $Y^{(k)}$  is a nonnegative matrix;
- $\sum_{i=k+1}^{\ell} T_{ij}^{(k)} \geq 0$  for  $j \in [k+1, t]$ ;
- $T_{ij}^{(k)} \leq 0$  for  $i \neq j$ ;
- $T_{ii}^{(k)} \geq 0$  for all  $i \in [k]$ .

Note that the last three properties imply that the lower right  $(\ell - k) \times (t - k)$  submatrix of  $T^{(k)}$  is a  $Z_+$ -matrix.

The properties clearly hold for  $k=0$ ; assume we have proved these for  $k-1$ . Consider the  $k$ -th iteration. If  $T_{kk}^{(k-1)} = 0$ , then no row operation is performed. In this case, we set  $T^{(k)} := T^{(k-1)}$  and  $Y^{(k)} := Y^{(k-1)}$ . We only need to verify that  $\sum_{i=k+1}^{\ell} T_{ij}^{(k)} \geq 0$  for  $j \in [k+1, t]$ . This follows from the induction hypotheses:  $\sum_{i=k}^{\ell} T_{ij}^{(k-1)} \geq 0$  and  $T_{kj}^{(k-1)} \leq 0$  for  $j \in [k+1, t]$ .

If  $T_{kk}^{(k-1)} > 0$ , then we multiply the  $k$ -th row by  $1/T_{kk}^{(k-1)}$  and add  $-T_{ik}^{(k-1)}/T_{kk}^{(k-1)}$  times the  $k$ -th row to the  $i$ -th row for all  $i \in [k+1, \ell]$ . By induction, these coefficients are all nonnegative. We update the transformation matrix  $Y^{(k)}$  accordingly, and, thus, it remains a nonnegative matrix. Consider now the  $j$ -th column of  $T^{(k)}$  for  $j \in [k+1, t]$ .

We have

$$\sum_{i=k+1}^{\ell} T_{ij}^{(k)} = \sum_{i=k+1}^{\ell} T_{ij}^{(k-1)} - T_{kj}^{(k-1)} \sum_{i=k+1}^{\ell} T_{ik}^{(k-1)} / T_{kk}^{(k-1)}. \quad (10)$$

The second induction hypothesis for  $j = k$  gives

$$T_{kk}^{(k-1)} + \sum_{i=k+1}^{\ell} T_{ik}^{(k-1)} \geq 0.$$

Rearranging, and using that  $T_{kk}^{(k-1)} > 0$ , we obtain

$$-\sum_{i=k+1}^{\ell} T_{ik}^{(k-1)} / T_{kk}^{(k-1)} \leq 1.$$

If we multiply this by  $T_{kj}^{(k-1)} \leq 0$ , we get

$$-T_{kj}^{(k-1)} \sum_{i=k+1}^{\ell} T_{ik}^{(k-1)} / T_{kk}^{(k-1)} \geq T_{kj}^{(k-1)}.$$

Substituting into (10), this yields

$$\sum_{i=k+1}^{\ell} T_{ij}^{(k)} \geq \sum_{i=k+1}^{\ell} T_{ij}^{(k-1)} + T_{kj}^{(k-1)} \geq 0,$$

again by the induction hypothesis.

For the last part, let  $T_{ik} < 0$  for  $k \in [t]$ ,  $i \in [k+1, \ell]$ . Note that  $T_{ik}^{(k-1)} \leq T_{ik}^{(k-2)} \leq \dots \leq T_{ik}^{(0)} < 0$ . The induction hypothesis gives  $\sum_{j=k}^{\ell} T_{jk}^{(k-1)} \geq 0$ , and, therefore,  $T_{kk}^{(k-1)} > 0$ . We set  $T_{kk}^{(k)} = 1$ , and this entry does not change in any later steps of the algorithm.  $\square$

### 7.3. Constructing the Approximate System

Let us now describe the construction of the M2VPI system  $\bar{M}p \leq \bar{\gamma}$  as in Theorem 5. We define a digraph  $([t], H)$  by adding an arc  $ij \in H$  if  $M_{ij} < 0$ . For each  $i \in [t]$ , we let  $D_i \subseteq [t]$  be the set of vertices reachable from  $i$  in the digraph  $([t], H)$ , and let  $d_i := |D_i|$ . We let  $M^{(i)}$  denote the  $d_i \times t$  submatrix of  $M$  comprising the rows  $M_j$  for  $j \in D_i$ . We partition  $[t]$  into three groups:

$$\begin{aligned} T_1 &:= \{i \in [t] : |\{j : ij \in H\}| \leq 1\}, \\ T_2 &:= \{i \in [t] \setminus T_1 : \text{rk}(M^{(i)}) = d_i\}, \\ T_3 &:= \{i \in [t] \setminus T_1 : \text{rk}(M^{(i)}) < d_i\}. \end{aligned}$$

If  $i \in T_1$ , then  $M_i$  has at most one positive and at most one negative entry; thus, we can keep the constraint  $M_i \bar{p} \leq \gamma_i$  unchanged. For  $i \in T_2 \cup T_3$ , for every outgoing arc  $ij \in H$ , we shall define a constraint in the form  $v^{(ij)\top} \bar{p} \geq \delta^{(ij)}$ . Further, for every  $i \in T_2$ , we shall add an additional constraint  $\bar{p}_i \leq \kappa_i$ .

The construction is somewhat technical, even though the underlying idea is relatively simple. For each  $ij \in H$ , we wish to obtain the constraint  $v^{(ij)\top} \bar{p} \geq \delta^{(ij)}$  such that  $\bar{p}_j$  has a positive coefficient,  $\bar{p}_i$  has a non-positive coefficient, and all other coefficients are zero. We wish to derive a valid constraint for  $(P_F)$  by taking

a nonnegative combination of constraints from  $M\bar{p} \geq -\lambda$ ; recall from Lemma 13 that these are valid for  $(P_F)$ . Lemma 14 shows that when we apply Gaussian elimination to a  $Z_+$ -matrix, then we only add rows with nonnegative coefficients. Hence, if we apply Gaussian elimination to the matrix  $M$  and apply the same operations to the right-hand side  $-\lambda$ , then we can derive valid constraints from  $M\bar{p} \geq -\lambda$ . In the construction that follows, we apply a permutation to a submatrix of  $M$ , wherein the penultimate step of Gaussian elimination produces a constraint of the desired form.

Let  $i \in T_2 \cup T_3$ , and  $d := d_i$ . For every  $ij \in H$ , let us define a permutation  $\sigma^{(ij)}$  of the set  $[t]$  as follows. We set  $\sigma(d-1) = j$ ,  $\sigma(d) = i$ , and fill the first  $d-2$  positions with the elements of  $D_i \setminus \{i\}$  in such a way that for any  $k \in D_i \setminus \{i\}$ , there is an edge  $\ell k \in H$  such that  $\sigma(k) < \sigma(\ell) \leq d$ . The final  $t-d$  positions contain the elements of  $[t] \setminus D_i$  in an arbitrary order. Let  $M^{(ij)} \in \mathbb{R}^{d \times t}$  denote the matrix obtained from  $M^{(i)}$  by applying the permutation  $\sigma^{(ij)}$  to the rows and the columns and deleting the last  $t-d$  rows. It is easy to see that  $M^{(ij)}$  is a  $Z_+$ -matrix.

**Example 3.** The picture shows the graph  $([t], H)$  for the system obtained from Figure 1. We have  $D_1 = D_2 = D_3 = \{1, 2, 3\}$ , and  $D_4 = D_5 = \{1, 2, 3, 4, 5\}$ . Now,

$$M^{(1)} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 5 & -1 & 0 & 0 \\ -1 & -2 & 3 & 0 & 0 \end{pmatrix}, \quad M^{(4)} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 5 & -1 & 0 & 0 \\ -1 & -2 & 3 & 0 & 0 \\ 0 & -2 & -1 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix},$$

and  $M^{(2)} = M^{(3)} = M^{(1)}$ ,  $M^{(5)} = M^{(4)}$ .

We get  $T_1 = \{5\}$ ,  $T_2 = \{1, 2, 3\}$ , and  $T_3 = \{4\}$ .

Let us apply Gaussian elimination as in Lemma 14 to  $M^{(ij)}$  to obtain an upper triangular matrix  $N^{(ij)} = Y^{(ij)} M^{(ij)}$ . Let  $\gamma^{(ij)}, \lambda^{(ij)} \in \mathbb{R}^d$  be the vectors obtained by permuting the components of  $\gamma$  and  $\lambda$  with  $\sigma^{(ij)}$  and removing the last  $t-d$  entries.

Let us set  $v^{(ij)}$  to be the  $(d-1)$ -st row  $N_{d-1}^{(ij)}$  with the inverse of the permutation  $\sigma^{(ij)}$  applied to its elements. (So that its  $i$ -th coordinate corresponds to  $\bar{p}_i$ .) Let

$$\delta^{(ij)} := -Y_{d-1}^{(ij)} \lambda^{(ij)}. \quad (11)$$

For  $i \in T_2$ , we add an additional constraint  $\bar{p}_i \leq \kappa^{(i)}$ . Let us pick an arbitrary  $ij \in H$ , and let

$$\kappa^{(i)} := Y_d^{(ij)} \gamma^{(ij)}. \quad (12)$$

It will be shown in Lemma 15 that this value is independent of the choice of the arc  $ij$ . The LP  $\bar{M}\bar{p} \leq \bar{\gamma}$  will

be the following system:

$$\begin{aligned} M_i \bar{p} &\leq \gamma_i & \forall i \in T_1, \\ \bar{p}_i &\leq \kappa^{(i)} & \forall i \in T_2, \\ v^{(ij)\top} \bar{p} &\geq \delta^{(ij)} & \forall ij \in H, i \in T_2 \cup T_3, \\ \bar{p} &\geq 0. \end{aligned} \quad (13)$$

#### 7.4. Proof of Correctness

We need one more claim before proving Theorem 5.

**Claim 1.** Let  $i \in [t]$  and let  $d := d_i$ .

**Example 4.** Continuing with the example, we have  $\gamma^\top = (2, 3, 3, 1, 1)$ , and  $\lambda^\top = (8, 7, 7, 9, 9)$ . Let us consider  $i=1, j=2$ . We use the permutation  $\sigma^{(12)} = (32145)$ , yielding

$$M^{(12)} = \begin{pmatrix} 3 & -2 & -1 & 0 & 0 \\ -1 & 5 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \end{pmatrix}, \quad \gamma^{(12)} = \begin{pmatrix} 3 \\ 3 \\ 2 \end{pmatrix}, \quad \lambda^{(12)} = \begin{pmatrix} 7 \\ 7 \\ 8 \end{pmatrix},$$

From Gaussian elimination, we get

$$N^{(12)} = \begin{pmatrix} 1 & -\frac{2}{3} & -\frac{1}{3} & 0 & 0 \\ 0 & 1 & -\frac{4}{13} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

$$Y^{(12)} \gamma^{(12)} = \begin{pmatrix} 1 \\ \frac{12}{13} \\ \frac{59}{15} \\ \frac{17}{15} \end{pmatrix}, \quad Y^{(12)} \lambda^{(12)} = \begin{pmatrix} \frac{7}{3} \\ \frac{28}{13} \\ \frac{181}{15} \end{pmatrix}.$$

This yields the constraint  $v^{(12)\top} \bar{p} \geq \delta^{(12)}$  for  $v^{(12)\top} = (-\frac{4}{13}, 1, 0, 0, 0)$ , and  $\delta^{(12)} = -\frac{28}{13}$ —that is,

$$-\frac{4}{13} \bar{p}_1 + \bar{p}_2 \geq -\frac{28}{13}.$$

Further, we can also use this to obtain  $\bar{p}_1 \leq \kappa^{(1)}$  for  $\kappa^{(1)} = \frac{59}{15}$ —that is,

$$\bar{p}_1 \leq \frac{59}{15}.$$

The system  $Q_F$  comprises the Constraint Set (7), and the following constraints:

$$\begin{aligned} \bar{p}_2 - \frac{4}{13} \bar{p}_1 &\geq -\frac{28}{13}, & \bar{p}_1 &\leq \frac{59}{15}, & (i=1, j=2) & & \bar{p}_2 &\geq -\frac{88}{15}, & (i=4, j=2) \\ \bar{p}_1 - \bar{p}_2 &\geq -\frac{31}{5}, & \bar{p}_2 &\leq \frac{32}{15}, & (i=2, j=1) & & \bar{p}_3 &\geq -\frac{154}{15}, & (i=4, j=3) \\ \bar{p}_3 - \frac{7}{13} \bar{p}_1 &\geq -\frac{49}{13}, & & & (i=1, j=3) & & \bar{p}_5 - \bar{p}_4 &\geq -9, & (i=4, j=5) \\ \bar{p}_1 - \frac{2}{3} \bar{p}_3 &\geq -\frac{47}{9}, & \bar{p}_3 &\leq \frac{61}{12}, & (i=3, j=1) & & \bar{p}_4 - \bar{p}_5 &\geq -31, & (i=5, j=4) \\ \bar{p}_3 - \bar{p}_2 &\geq -\frac{22}{5}, & & & (i=2, j=3) & & & & \\ \bar{p}_2 - \frac{1}{3} \bar{p}_3 &\geq -\frac{22}{9}, & & & (i=3, j=2) & & & & \end{aligned}$$

i. For any  $ij \in H$ ,  $N_{kk}^{(ij)} = 1$  for all  $k \in [d-1]$ , and  $N_{k\ell}^{(ij)} = 0$  for all  $k \in [d]$ ,  $\ell \in [d+1, t]$ .

ii. If  $i \in T_2$ , then  $N_{dd}^{(ij)} = 1$ , and if  $i \in T_3$ , then  $N_{dd}^{(ij)} = 0$ .

iii. If  $i \in T_3$ , then  $M_i$  can be written as a linear combination of the vectors  $\{M_h : h \in D_i \setminus \{i\}\}$ .

**Proof.**

i. In the definition of the permutation  $\sigma^{(ij)}$ , it was required that for any  $k \in [d-1]$ , there is an entry  $M_{k\ell}^{(ij)} < 0$  for some  $\ell$  with  $\sigma(k) < \sigma(\ell) \leq d$ . The last part of Lemma 14 guarantees that  $N_{kk}^{(ij)} = 1$  for all  $k \in [d-1]$ . Further, we note that according to the definition of the set  $D_i$ , we have  $M_{k\ell}^{(ij)} = 0$  for all  $k \in [d]$ ,  $\ell \in [d+1, t]$ . Thus, the last  $t-d$  entries of every row in  $M^{(ij)}$  are zeros and therefore also in  $N^{(ij)}$ .

ii. Together with the fact the  $N^{(ij)}$  is an upper triangular matrix, we see that the only nonzero entry of the  $d$ -th row  $N_d^{(ij)}$  is the diagonal entry  $N_{dd}^{(ij)} \in \{0, 1\}$ . Also note that  $\text{rk}(N^{(ij)}) = \text{rk}(M^{(ij)}) = \text{rk}(M^{(i)})$ . If  $i \in T_2$ , then we must have  $\text{rk}(N^{(ij)}) = d$ , and, therefore,  $N_{dd}^{(ij)} = 1$ , as otherwise  $N_d^{(ij)} = 0$ . On the other hand, if  $i \in T_3$ , then  $N_{dd}^{(ij)} = 0$ , as otherwise,  $N^{(ij)}$  would be an upper triangular matrix with the first  $d$  diagonal entries equal to one, contradicting  $\text{rk}(N^{(ij)}) < d$ .

iii. We claim that  $\text{rk}(M_h : h \in D_i \setminus \{i\}) = d-1$ . This implies the statement because we assumed that  $\text{rk}(M^{(i)}) = \text{rk}(M_h : h \in D_i) = d-1$ . Note that the first  $d-1$  rows of  $M^{(ij)}$  are permutations of the vectors  $M_h$  for  $h \in D_i \setminus \{i\}$  and, hence, have the same rank. The rank is preserved during Gaussian elimination. Therefore,

$$\begin{aligned} \text{rk}(M_h : h \in D_i \setminus \{i\}) &= \text{rk}(M_h^{(ij)} : h \in [d-1]) \\ &= \text{rk}(N_h^{(ij)} : h \in [d-1]) = d-1. \end{aligned}$$

The last equality follows because  $N^{(ij)}$  is an upper triangular matrix with the first  $d-1$  diagonal entries being one, according to part (i).  $\square$

#### Proof of Theorem 5.

**Form of the constraints.** First, let us show that the system  $\bar{M} \bar{p} \leq \bar{\gamma}$  given in (13) is an M2VPI system. This is clearly true for the constraints for  $i \in T_1$  and for  $i \in T_2$ . Consider now the constraints  $v^{(ij)\top} \bar{p} \geq \delta^{(ij)}$ . The vector  $v^{(ij)}$  was obtained as the appropriate permutation of the  $(d-1)$ -st row of the matrix  $N^{(ij)}$ . Using that  $N^{(ij)}$  is an upper triangular matrix and Claim 1(i), this row may contain nonzero entries only in positions  $d-1$  and  $d$ . Further,  $v_j^{(ij)} = N_{(d-1)(d-1)}^{(ij)} = 1$ , and  $v_i^{(ij)} = N_{(d-1)d}^{(ij)} \leq 0$ , as it is an off-diagonal entry.

**Containment of  $P_M$ .** We show that every  $\bar{p}$  satisfying  $\bar{M} \bar{p} \leq \bar{\gamma}$  is also feasible to (13). For  $i \in T_1$ , the constraint  $M_i \bar{p} \leq \gamma_i$  is identical to the  $i$ -th constraint in  $P_M$ .

For  $i \in T_2$ , let  $ij \in H$  be the edge used in the definition of  $\kappa^{(i)}$ . According to Claim 1, the row  $N_d^{(ij)}$  has a



single nonzero entry  $N_{dd}^{(ij)} = 1$ . Lemma 14 guarantees that the coefficient matrix  $Y^{(ij)}$  is nonnegative. Therefore, the constraint  $\bar{p}_i \leq \kappa^{(i)}$  can be obtained as a nonnegative combination of the constraint set  $M\bar{p} \leq \gamma$ , by multiplying  $M_h\bar{p} \leq \gamma_h$  for  $h \in D_i$  by  $Y_{d\sigma^{(ij)}(h)}^{(ij)}$ .

The validity of the constraints  $v^{(ij)\top}\bar{p} \geq \delta^{(ij)}$  follows similarly. Recall from Lemma 13 that  $M\bar{p} \geq -\lambda$  is valid for  $\bar{p}$ . The constraint  $v^{(ij)\top}\bar{p} \geq \delta^{(ij)}$  is obtained by taking a nonnegative combination of the inequalities  $M\bar{p} \geq -\lambda$  combining  $M_h\bar{p} \geq -\lambda_h$  for  $h \in D_i$  with the nonnegative coefficient  $Y_{(d-1)\sigma^{(ij)}(h)}^{(ij)}$ . Hence, all these inequalities are valid for  $\bar{p}$ .

**Approximate reverse containment.** We next show that if  $\bar{p}$  is feasible to (13), then  $\bar{p}$  is feasible to  $B^2P_M$ —that is,  $M\bar{p} \leq B^2\gamma$ . Clearly, for  $i \in T_1$ ,  $M_i\bar{p} \leq \gamma_i \leq B^2\gamma_i$ . The more difficult part is to show the validity of  $M_i\bar{p} \leq B^2\gamma_i$  for  $i \in T_2 \cup T_3$ .

For  $i \in T_3$ , we show that the constraints

$$v^{(ij)\top}\bar{p} \geq \delta^{(ij)} \quad \forall j: ij \in H$$

together imply  $M_i\bar{p} \leq B^2\gamma_i$ . For  $i \in T_2$ , we will also make use of the additional constraint  $\bar{p}_i \leq \kappa^{(i)}$  to derive  $M_i\bar{p} \leq B^2\gamma_i$ . The following technical lemma will be needed.

**Lemma 15.** Consider any  $i \in T_2 \cup T_3$ .

- i. There is a unique vector  $q^{(i)} \in \mathbb{R}_+^t$  such that  $M_\ell q^{(i)} = 0$  for all  $\ell \in D_i \setminus \{i\}$ ,  $q_i^{(i)} = 1$ , and  $q_\ell^{(i)} = 0$  for  $\ell \in [t] \setminus D_i$ .
- ii. For any  $ij \in H$ ,  $v_j^{(ij)} = 1$  and  $v_i^{(ij)} = -q_j^{(i)}$ .
- iii. If  $i \in T_2$ , then there exists a vector  $r^{(i)} \in \mathbb{R}_+^t$  with  $Mr^{(i)} \leq \gamma$ ,  $M_\ell r^{(i)} = \gamma_\ell$  for all  $\ell \in D_i$ , and  $r_i^{(i)} = \kappa^{(i)}$ .
- iv. If  $i \in T_3$ , then there exists a vector  $r^{(i)} \in \mathbb{R}_+^t$  with  $Mr^{(i)} \leq \gamma$ , and  $M_\ell r^{(i)} = \gamma_\ell$  for all  $\ell \in D_i \setminus \{i\}$ .

**Proof.** As before, we let  $d = d_i$ . For part (i), let us apply Gaussian elimination, as in Lemma 14, to the matrix  $M^{(ij)}$  for any  $ij \in H$ . Note that  $M_\ell^{(ij)}\hat{q} = 0$  for  $\ell \in [d-1]$  is equivalent to  $M_\ell q = 0$  for all  $\ell \in D_i \setminus \{i\}$ , where  $\hat{q}$  is the vector obtained from  $q$  by permuting the elements by  $\sigma^{(ij)}$ . The system of linear equations  $M_\ell^{(ij)}\hat{q} = 0$  for  $\ell \in [d-1]$  is turned into the system  $N_\ell^{(ij)}\hat{q} = 0$  for  $\ell \in [d-1]$  using the elimination.

From Claim 1, we have that  $N_{kk}^{(ij)} = 1$  for all  $k \in [d-1]$ , and all off-diagonal elements of the upper triangular matrix  $N^{(ij)}$  are nonpositive. Hence, if we set  $\hat{q}_k = 0$  for all  $k \in [d+1, t]$  and  $\hat{q}_d = 1$ , then the system  $N_\ell^{(ij)}\hat{q} = 0$  for  $\ell \in [d-1]$  has a unique nonnegative solution. Applying the inverse of  $\sigma^{(ij)}$  to  $\hat{q}$ , we obtain the desired vector  $q^{(i)}$ .

For part (ii), recall that  $v_i^{(ij)}$  is obtained from  $N_{d-1}^{(ij)}$  using the inverse of the permutation  $\sigma^{(ij)}$ . We have already verified that the only possible nonzero entries in  $N_{d-1}^{(ij)}$  are the  $(d-1)$ -st and  $d$ -th components and that  $v_j^{(ij)} = N_{(d-1)(d-1)}^{(ij)} = 1$ . Because  $N_{d-1}^{(ij)}\hat{q} = 0$  and  $\hat{q}_d = q_i^{(i)} = 1$ ,

we must have  $\hat{q}_{d-1} + N_{(d-1)d}^{(ij)} = 0$ , and thus  $N_{(d-1)d}^{(ij)} = -\hat{q}_{d-1}$ ; after permuting, this gives  $v_i^{(ij)} = -q_j^{(i)}$ .

Let us now turn to part (iii). Let  $i \in T_2$ , and let us fix an arbitrary  $ij \in H$ . The system  $M_\ell r = \gamma_\ell$  for all  $\ell \in D_i$  is equivalent to  $M^{(ij)}\hat{r} = \gamma^{(ij)}$ , where  $\hat{r}$  is obtained from  $r$  by the permutation  $\sigma^{(ij)}$ . After Gaussian elimination, we obtain the equivalent system

$$N^{(ij)}\hat{r} = Y^{(ij)}\gamma^{(ij)}.$$

According to Claim 1,  $N_{\ell\ell}^{(ij)} = 1$  for all  $\ell \in [d]$ . This system has a unique solution  $\hat{r}$  with  $\hat{r}_\ell = 0$  for  $d+1 \leq \ell \leq t$ ; note also that  $\hat{r}_d = Y_d^{(ij)}\gamma^{(ij)} = \kappa^{(i)}$ . Because the right-hand side is nonnegative, and  $N^{(ij)}$  is an upper triangular matrix with all off-diagonal entries being nonpositive, we see that  $\hat{r}_\ell \geq 0$  for all  $\ell \in [d]$ . Let  $r^{(i)}$  be the vector obtained from  $\hat{r}$  by applying the inverse permutation of  $\sigma^{(ij)}$ . It remains to show that  $M_\ell r^{(i)} \leq \gamma_\ell$  holds for all  $\ell \in [t] \setminus D_i$ . The only positive coefficient in  $M_\ell$  can be  $M_{\ell\ell}$ . However, because  $r_\ell^{(i)} = 0$ , it follows that  $M_\ell r^{(i)} \leq 0$  for these values of  $\ell$ .

The proof of part (iv) is similar. Let  $i \in T_3$ , and let us fix an arbitrary  $ij \in H$ . The system  $M_\ell r = \gamma_\ell$  for all  $\ell \in D_i \setminus \{i\}$  is equivalent to  $M_\ell^{(ij)}\hat{r} = \gamma_\ell^{(ij)}$  for  $\ell \in [d-1]$ , where  $\hat{r}$  is obtained by applying the permutation  $\sigma^{(ij)}$  to the vector  $r$ . After Gaussian elimination, we obtain the equivalent system

$$N_\ell^{(ij)}\hat{r} = Y_\ell^{(ij)}\gamma^{(ij)} \quad \forall \ell \in [d-1],$$

with  $N_{\ell\ell}^{(ij)} = 1$  for all  $\ell \in [d-1]$ . As in part (iii), we see that there is a unique solution with  $\hat{r}_\ell = 0$  for  $d \leq \ell \leq t$ , and this solution satisfies  $\hat{r}_\ell \geq 0$  for all  $\ell \in [d-1]$ . We obtain  $r^{(i)}$  by applying the inverse of  $\sigma^{(ij)}$  to  $\hat{r}$ . It follows as above that  $M_\ell r^{(i)} \leq 0$  for all  $\ell \in [t] \setminus D_i$ . The same argument also gives  $M_i r^{(i)} \leq 0$ , because  $r_i^{(i)} = 0$ . Hence,  $Mr^{(i)} \leq \gamma$  holds.  $\square$

Assume now that  $i \in T_2 \cup T_3$ . We show that  $M_i\bar{p} \leq B^2\gamma_i$ . Let  $q := q^{(i)}$  as in Lemma 15(i). By part (ii) of the same lemma, and substituting the Definition (11) of  $\delta^{(ij)}$ , the constraints can be written as

$$\bar{p}_j - q_j\bar{p}_i \geq -Y_{d-1}^{(ij)}\lambda^{(ij)} \quad \forall j: ij \in H.$$

Note that  $\lambda_\ell^{(ij)} \leq (B-1)\gamma_\ell^{(ij)}$  for all  $\ell \in [d]$  by the definition of  $B$ , and  $Y_{d-1}^{(ij)} \geq 0$ . Therefore, these constraints imply

$$\bar{p}_j - q_j\bar{p}_i \geq -(B-1)Y_{d-1}^{(ij)}\gamma^{(ij)} \quad \forall j: ij \in H.$$

Recall that  $M_{ij} < 0$  if and only if  $ij \in H$ . Let us multiply the inequality for every  $j \neq i$  by  $M_{ij} \leq 0$  and add up these inequalities. We obtain

$$\sum_{j:ij \in H} M_{ij}\bar{p}_j - \left( \sum_{j:ij \in H} M_{ij}q_j \right) \bar{p}_i \leq -(B-1) \sum_{j:ij \in H} M_{ij}Y_{d-1}^{(ij)}\gamma^{(ij)}. \quad (14)$$

For the rest of the proof, we distinguish the cases  $i \in T_3$  and  $i \in T_2$ .

**Case  $i \in T_3$ .** Because  $M_h q = 0$  for all  $h \in D_i \setminus \{i\}$ , Claim 1(iii) implies  $M_i q = 0$ . Substituting  $q_i = 1$ , we see that  $M_{ii} = -\sum_{j:ij \in H} M_{ij} q_j$ . With  $\eta_i := -\sum_{j:ij \in H} M_{ij} Y_{d-1}^{(ij)} \gamma^{(ij)}$ , (14) can be written as

$$M_{ii} \bar{p}_i + \sum_{j:ij \in H} M_{ij} \bar{p}_j \leq (B-1)\eta_i. \quad (15)$$

The left-hand side is  $M_i \bar{p}$ . We next show  $\eta_i \leq \lambda_i$ , which together with  $\lambda_i \leq (B-1)\gamma_i$  yields  $M_i \bar{p} \leq (B-1)^2 \gamma_i$ .

To see  $\eta_i \leq \lambda_i$ , we make use of the vector  $r = r^{(i)}$  as in Lemma 15(iv). Let  $\hat{r}$  denote the permutation  $\sigma^{(ij)}$  applied to  $r$ . Because  $M_\ell^{(ij)} \hat{r} = \gamma_\ell$  is valid for all  $\ell \in [d-1]$ , we have  $N_{d-1}^{(ij)} \hat{r} = Y_{d-1}^{(ij)} \gamma^{(ij)}$ . This can be written as

$$r_j - q_j r_i = Y_{d-1}^{(ij)} \gamma^{(ij)} \quad \forall ij \in H. \quad (16)$$

Summing up these equalities after multiplying the  $j$ -th one by  $M_{ij} < 0$ , we see, as above, that

$$M_i r = -\eta_i.$$

Because  $M r \leq \gamma$ , from Lemma 13, we have  $-\eta_i = M_i r \geq -\lambda_i$ , and, therefore,  $\eta_i \leq \lambda_i$  as needed.

**Case  $i \in T_2$ .** The coefficient of  $\bar{p}_i$  in (14) equals  $M_{ii} - M_i q$ . In contrast with the previous case,  $M_i q$  is not necessarily zero. We claim that  $M_i q \geq 0$ . To see this, note that  $\sum_{h \in D_i} M_h q \geq 0$  because  $\sum_{h \in D_i} M_h \geq 0$  from the  $Z_+$ -property and  $q \geq 0$ ; further,  $M_h q = 0$  for  $h \in D_i \setminus \{i\}$ . Let us further add to (14)  $M_i q$  times the inequality  $\bar{p}_i \leq \kappa^{(i)}$ . Thus, we obtain

$$M_i \bar{p} \leq (B-1)\eta_i + \kappa^{(i)} M_i q. \quad (17)$$

Let  $r = r^{(i)}$  as in Lemma 15(iii). As for  $i \in T_3$ , the Equations (16) hold. Adding up these equations multiplied by  $M_{ij}$ , and further adding  $M_i q$  times the equality  $r_i = \kappa^{(i)}$ , we obtain

$$M_i r = -\eta_i + \kappa^{(i)} M_i q.$$

On the other hand, we know that  $M_i r = \gamma_i$ . Thus,  $\gamma_i = -\eta_i + \kappa^{(i)} M_i q$ , yielding  $\eta_i \leq \kappa^{(i)} M_i q$ . Consequently, from (17), we obtain

$$M_i \bar{p} \leq B \kappa^{(i)} M_i q. \quad (18)$$

The next claim completes the proof of  $M_i \bar{p} \leq B^2 \gamma_i$ .

**Claim 2.**  $\kappa^{(i)} M_i q \leq B \gamma_i$ .

**Proof.** For  $ij \in H$  used in defining  $\kappa^{(i)}$ , let  $\hat{q}$  be the vector obtained from  $q$  by applying the permutation  $\sigma^{(ij)}$ . We first show that  $M_i q = 1/Y_{dd}^{(ij)}$ . To see this, first note that  $N_d^{(ij)} \hat{q} = 1$ , because  $N_{dd}^{(ij)} = 1$ ,  $\hat{q}_d = 1$ , and all other entries are zero. We have

$$1 = N_d^{(ij)} \hat{q} = Y_d^{(ij)} (M^{(ij)} \hat{q}) = Y_{dd}^{(ij)} (M_d^{(ij)} \hat{q}) = Y_{dd}^{(ij)} (M_i q).$$

The third equality follows because  $M_\ell^{(ij)} \hat{q} = 0$  for  $\ell \in [d-1]$ . Using the Definition (12) of  $\kappa^{(i)}$ , we have

that

$$\kappa^{(i)} M_i q = \sum_{h=1}^d \frac{Y_{dh}^{(ij)}}{Y_{dd}^{(ij)}} \gamma_h^{(ij)}. \quad (19)$$

Let us now show that  $Y_{dh}^{(ij)} \leq Y_{dd}^{(ij)}$  for all  $h \in [d]$ . Let us pick  $h$  such that it maximizes  $Y_{dh}^{(ij)}$  over  $h \in [d]$  and select the largest  $h$  where the maximum is taken. For a contradiction, assume that  $h < d$ . Because  $N^{(ij)}$  is an upper triangular matrix, we have

$$0 = N_{dh}^{(ij)} = \sum_{\ell=1}^d Y_{d\ell}^{(ij)} M_{\ell h}^{(ij)}.$$

Using that  $M^{(ij)}$  is a  $Z_+$ -matrix, we have  $M_{hh}^{(ij)} \geq 0$ ,  $M_{\ell h}^{(ij)} \leq 0$  for  $\ell \neq h$ , and  $\sum_{\ell=1}^d M_{\ell h}^{(ij)} \geq 0$ . Using the maximality of  $Y_{dh}^{(ij)}$ , we see that equality can only hold if  $\sum_{\ell=1}^d M_{\ell h}^{(ij)} = 0$ , and  $Y_{d\ell}^{(ij)} = Y_{dh}^{(ij)}$  whenever  $M_{\ell h}^{(ij)} < 0$ . This contradicts the maximal choice of  $h$ , because the permutation  $\sigma^{(ij)}$  was chosen so that there exists an  $\ell > h$  with  $M_{\ell h}^{(ij)} < 0$  whenever  $h < d$ . Thus, we can conclude that  $Y_{dh}^{(ij)} \leq Y_{dd}^{(ij)}$  for all  $h \in [d]$ . From (19), it follows that

$$\kappa^{(i)} M_i q \leq \sum_{h=1}^d \gamma_h^{(ij)} = \sum_{h \in D_i} \gamma_h \leq B \gamma_i,$$

using the definition of  $B$ .  $\square$

**7.4.1. Running Time and Encoding Length.** We can obtain the constraints in (13) by running Gaussian elimination for each  $ij \in H$ ; this takes altogether  $O(\ell(\min\{k, t\})^3)$  time. We need to show that the encoding size of  $\bar{M}$  and  $\bar{b}$  are polynomially bounded in the encoding size of  $M$  and  $b$ . This easily follows because the constraints are obtained by Gaussian elimination; we refer to Edmonds (1967) for strong polynomiality of Gaussian elimination.

The proof of Theorem 5 is now complete.  $\square$

## 8. Conclusions

We have given a strongly polynomial algorithm for computing an exact equilibrium in linear exchange markets. We use a variant of the Duan-Mehlhorn algorithm as a subroutine in a framework that repeatedly identifies revealed arcs. Before each iteration of this subroutine, we use another method to find a good starting solution for the current set of revealed arcs. The best solution here corresponds to the optimal solution of a linear program. Whereas no strongly polynomial algorithm is known for an LP of this form, we presented a strongly polynomial approximation by constructing a second LP.

The goal of this paper was to get a better understanding of theoretical solvability of this important

practical problem. For practical purposes, the approximate algorithms mentioned in Section 1 may suffice; in particular, the simple and efficient  $O(\frac{n}{\epsilon}(m + n \log n))$  algorithm by Ghyasvand and Orlin (2012) would be a natural choice. Although we have not implemented our algorithm, we believe that it should be much better in practice than the worst-case estimate  $O(n^{10} \log^2 n)$ . We are not aware of previous computational studies on exchange-market algorithms; we leave the comparison of our algorithm to Ye (2008) and Duan et al. (2016) for future work.

From a theoretical perspective, it could be worth exploring whether this approach extends further. An immediate question is to see if one can use such an approach to obtain a  $\epsilon$ -approximation of the LP in strongly polynomial time for every  $\epsilon > 0$ . Further, such a method could be potentially useful for a broader class of LPs; a natural candidate would be systems of the form  $A^T x \leq c$  for a pre-Leontief matrix  $A$  (see Cottle and Veinott 1972), a class where a pointwise maximal solution exists, but no strongly polynomial algorithm is known.

Our approach was specific to the market equilibrium problem. The method of identifying revealed arc sets originates from Végh (2016). This result was applicable not only for the linear Fisher market model, but, more generally, for minimum-cost flow problems with separable convex objectives satisfying certain assumptions. It would be desirable to extend the current approach to a broader class of convex programs that include the formulation in Devanur et al. (2016).

### Acknowledgments

The authors are grateful to Kurt Mehlhorn, Vijay Vazirani, and Richard Cole for many interesting discussions on this problem.

### Appendix: Proof of Lemma 9

We will use the following simple lemmas to prove the result.

**Lemma A.1** (Duan et al. 2016). *Let  $r = (r_1, \dots, r_n)$  and  $r' = (r'_1, \dots, r'_n)$  be nonnegative vectors. Let  $k \in [1, n]$  be such that  $r'_i \geq r'_j$  for  $i \leq k < j$ . Suppose that  $\delta_i = r_i - r'_i \geq 0$  for  $i \leq k$  and  $\delta_j = r'_j - r_j \geq 0$  for  $j > k$ . Let  $D = \min_{i \leq k} r_i - \max_{j > k} r'_j$ , and let  $\Delta = \sum_{i \leq k} \delta_i$ . If  $\Delta \geq \sum_{j > k} \delta_j$ ,*

$$\|r'\|_2^2 \leq \|r\|_2^2 - D\Delta.$$

**Lemma A.2** (Duan and Mehlhorn 2015, Duan et al. 2016). *Given  $n$  numbers  $a_1 \geq a_2 \geq \dots \geq a_n \geq 0$ . Let  $l$  be minimal such that  $a_l/a_{l+1} \leq 1 + 1/n$ . Let  $l = n$  if there is no such  $l$ . Then  $(e \cdot n)a_l \geq \|a\|_1$  and  $a_i \leq e \cdot a_l$ ,  $\forall i$ .*

The proof of the following lemma is an adaptation of the proof given in Duan et al. (2016).

**Lemma 9.** *Let  $f$  be a balanced flow at the beginning of a phase, and let  $(p', f')$  be the prices and flow at the end of the phase. Then,*

- i.  $\prod_{j=1}^n p'_j \geq \left(1 + \frac{1}{56e^2 n^3}\right) \prod_{j=1}^n p_j$  in a price-rise phase, and
- ii.  $\|c(p', f')\|_2 \leq \|c(p, f)\|_2 / \left(1 + \frac{1}{56e^2 n^3}\right)$  in a balancing phase.

**Proof.** Let us denote  $C = 56e^2$ , and let  $x_{\max} = 1 + \frac{1}{Cn^3}$ . Recall that price-rise phases correspond to Event 3. Because no good leaves  $\Gamma(S)$  during a phase and prices monotonically increase, Event 3 implies that price of at least one good has increased by  $x_{\max}$ . This proves the first part.

For the second part, the phase terminates either due to Event 2 or breaking from the inner loop (in line 2). For the latter case, there is a way to adjust the flow so that the surplus of an agent  $i \in S$  becomes equal to either zero if  $S = A$  or the surplus of an agent  $i' \notin S$ . Furthermore, the flow adjustment maintains that the surplus of each agent  $i \in S$  increases by a factor of at most  $x_{\max}$ , surpluses of agents outside  $S$  does not decrease, and the total surplus does not increase. This is a similar situation like Event 2, so we can prove this case by the proof of Event 2 case.

Let  $f''$  be the intermediate flow just before computing a balanced flow in line 2. If Event 2 occurs, then it implies that  $p'_j \leq x_{\max} p_j$ ,  $\forall j$ . Because  $p_j$  and  $f_{ij}$  are increased by the same factor  $x$ ,  $\forall i \in S$ ,  $\forall j \in \Gamma(S)$ , and the flow update in line 2 can only decrease the surplus of agents in  $S$ , we have  $c_i(p', f'') \leq x_{\max} c_i(p, f)$ ,  $\forall i \in S$ . There are two cases: Either the surplus of an agent  $i \in S$  becomes zero if  $S = A$ , or it becomes equal to surplus of an agent  $i' \notin S$ .

For the first case, we have  $c_i(p', f'') = 0$  and  $c_j(p', f'') \leq x_{\max} c_j(p, f)$ ,  $\forall j \in S$ . This implies that

$$\|c(p', f'')\|_2^2 \leq x_{\max}^2 \|c(p, f)\|_2^2 - c_i(p, f)^2.$$

Lemma A.2 implies that  $c_i(p, f) \geq \|c(p, f)\|_1 / (e \cdot n)$ . Using this, we get

$$\|c(p', f'')\|_2^2 \leq \|c(p, f)\|_2^2 \left( \left(1 + \frac{1}{Cn^3}\right)^2 - \frac{1}{e^2 n^2} \right).$$

Simplifying above using  $C = 56e^2$ , we get

$$\begin{aligned} \|c(p', f'')\|_2^2 &\leq \left(1 + \frac{3}{Cn^3} - \frac{56}{Cn^3}\right) \|c(p, f)\|_2^2 \\ &< \left(1 - \frac{4}{Cn^3}\right) \|c(p, f)\|_2^2. \end{aligned}$$

Further, we obtain

$$\begin{aligned} \|c(p', f'')\|_2 &\leq \sqrt{\left(1 - \frac{4}{Cn^3}\right)} \|c(p, f)\|_2 \leq \left(1 - \frac{2}{Cn^3}\right) \|c(p, f)\|_2 \\ &\leq \|c(p, f)\|_2 / \left(1 + \frac{1}{Cn^3}\right). \end{aligned}$$

Because  $\|c(p', f')\|_2 \leq \|c(p', f'')\|_2$ , we get

$$\|c(p', f')\|_2 \leq \|c(p, f)\|_2 / \left(1 + \frac{1}{Cn^3}\right).$$

For the second case,  $\min_{i \in S} c_i(p', f'') = \max_{i \notin S} c_i(p', f'')$ . There are two types of agents in  $S$ :  $i$  such that  $g_i \in \Gamma(S)$  (type 1) and  $i$  such that  $g_i \notin \Gamma(S)$  (type 2). The price and flow update in line 2 increases the surpluses of type 1 agents and decreases the surpluses of type 2 agents. Because the price and flow are updated by the same factor  $x$ , the surpluses of type 1 agents increase by the same factor  $x$ .

Because the surpluses of agents outside  $S$  does not decrease, and the total surplus monotonically decreases, the



total increase in the surpluses of agents outside  $S$  is at most the total decrease in the surpluses of type 2 agents. For simplicity, let  $w_1, \dots, w_\ell$  and  $u_1, \dots, u_k$  denote the surpluses of type 1 and type 2 agents at  $(p, f)$ , respectively. Similarly, let  $v_1, \dots, v_o$  denote the surpluses of agents outside  $S$  at  $(p, f)$ . Clearly, we have  $l + k + o = n$ . Define  $\bar{w} = \min_i w_i, \bar{u} = \min_i u_i, \bar{v} = \max_j v_j$ . Let  $R = \min\{\bar{u}, \bar{w}\}$ . From the definition of  $S$ , we have  $R > (1 + 1/n)\bar{v}$ . At  $(p', f')$ , let  $u'_1, \dots, u'_k, w'_1, \dots, w'_\ell, v'_1, \dots, v'_o$  are the surpluses of type 1, type 2, and agents outside  $S$ , respectively. From the above discussion, we have

$$w'_i \leq x_{\max} w_i, \forall i, \text{ and } u'_i \leq u_i, \forall i \text{ and } v'_i \geq v_i, \forall i.$$

Let  $u'_i = u_i - \delta_i, \forall i$  and  $v'_j = v_j + \delta'_j, \forall j$ , where  $\delta_i, \delta'_j \geq 0, \forall i, j$ . Further, we have  $w'_i \geq v'_i, \forall i, u'_i \geq v'_i, \forall i$ , and  $\sum_i \delta_i \geq \sum_j \delta'_j$ . Because  $R - \bar{v} \geq R/(n+1)$  and Event 2 occurred, we have  $\sum_i \delta_i \geq R/(2(n+1))$ . Using Lemma A.1, we get

$$\|u'\|_2^2 + \|v'\|_2^2 \leq \|u\|_2^2 + \|v\|_2^2 - \frac{R^2}{2(n+1)^2}.$$

Recall that  $w'_i \leq x_{\max} w_i, \forall i$ , and let us use the trivial upper bound  $\|w\|_2^2 \leq \|c(p, f)\|_2^2$ . Together with the above inequality, we obtain

$$\begin{aligned} \|c(p', f')\|_2^2 &= \|w'\|_2^2 + \|u'\|_2^2 + \|v'\|_2^2 \\ &\leq x_{\max}^2 \|w\|_2^2 + \|u\|_2^2 + \|v\|_2^2 - \frac{R^2}{2(n+1)^2} \\ &\leq \left(1 + \frac{1}{Cn^3}\right)^2 \|c(p, f)\|_2^2 - \frac{R^2}{2(n+1)^2}. \end{aligned}$$

From Lemma A.2, we have  $\|c(p, f)\|_2^2 \leq ne^2 R^2$ . Hence,

$$\begin{aligned} \|c(p', f')\|_2^2 &\leq \|c(p, f)\|_2^2 \left( \left(1 + \frac{1}{Cn^3}\right)^2 - \frac{1}{2e^2 n(n+1)^2} \right) \\ &\leq \|c(p, f)\|_2^2 / \left(1 + \frac{4}{Cn^3}\right), \end{aligned}$$

where the last inequality used that  $C = 56e^2$ . We obtain

$$\|c(p', f')\|_2 \leq \frac{\|c(p, f)\|_2}{\sqrt{1 + \frac{4}{Cn^3}}} \leq \frac{\|c(p, f)\|_2}{1 + \frac{1}{Cn^3}}.$$

Finally, because  $\|c(p', f')\|_2 \leq \|c(p', f'')\|_2$ , we get  $\|c(p', f')\|_2 \leq \|c(p, f)\|_2 / (1 + \frac{1}{Cn^3})$ .  $\square$

## References

- Adler I, Cosares S (1991) A strongly polynomial algorithm for a special class of linear programs. *Oper. Res.* 39(6):955–960.
- Arrow KJ, Debreu G (1954) Existence of an equilibrium for a competitive economy. *Econometrica* 22(3):265–290.
- Brainard W, Scarf H (2000) How to compute equilibrium prices in 1891. Cowles Foundation Discussion Paper 1270, Cowles Foundation, New Haven, CT.
- Codenotti B, Pemmaraju S, Varadarajan K (2004) The computation of market equilibria. *ACM SIGACT News* 35(4):23–37.
- Cohen E, Megiddo N (1994) Improved algorithms for linear inequalities with two variables per inequality. *SIAM J. Comput.* 23(6):1313–1347.
- Cornet B (1989) Linear exchange economies. Technical report, Cahier Eco-Math, Université de Paris, Paris.
- Cottle RW, Veinott AF (1972) Polyhedral sets having a least element. *Math. Program.* 3(1):238–249.
- Darwish O, Mehlhorn K (2016) Improved balanced flow computation using parametric flow. *Inform. Process. Lett.* 116(9):560–563.
- Devanur NR, Vazirani VV (2003) An improved approximation scheme for computing Arrow-Debreu prices for the linear case. Pandya PK, Radhakrishnan J, eds. *FST TCS 2003 Foundation Software Tech. Theoret. Comput. Sci., Lecture Notes in Computer Science*, vol. 2914 (Springer, Berlin), 149–155.
- Devanur NR, Garg J, Végh LA (2016) A rational convex program for linear Arrow-Debreu markets. *ACM Trans. Econom. Comput.* 5(1):6.
- Devanur NR, Papadimitriou CH, Saberi A, Vazirani VV (2008) Market equilibrium via a primal-dual algorithm for a convex program. *J. ACM* 55(5):22.
- Duan R, Mehlhorn K (2015) A combinatorial polynomial algorithm for the linear Arrow-Debreu market. *Inform. Comput.* 243:112–132.
- Duan R, Garg J, Mehlhorn K (2016) An improved combinatorial polynomial algorithm for the linear Arrow-Debreu market. Kraughamer R, ed. *SODA'16 Proc. 27th Annu. ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), 90–106.
- Eaves BC (1976) A finite algorithm for the linear exchange model. *J. Math. Econom.* 3:197–203.
- Edmonds J (1967) Systems of distinct representatives and linear algebra. *J. Res. National Bureau Standards B* 71:241–245.
- Eisenberg E, Gale D (1959) Consensus of subjective probabilities: The Pari-Mutuel method. *Ann. Math. Statist.* 30(1):165–168.
- Gale D (1976) The linear exchange model. *J. Math. Econom.* 3(2):205–209.
- Garg R, Kapoor S (2006) Auction algorithms for market equilibrium. *Math. Oper. Res.* 31(4):714–729.
- Garg J, Végh LA (2019) A strongly polynomial algorithm for linear exchange markets. *STOC 2019 Proc. 51st Annu. ACM SIGACT Sympos. Theory Comput.* (Association for Computing Machinery, New York), 54–65.
- Garg J, Mehta R, Vazirani VV, Yazdanbod S (2017) Settling the complexity of Leontief and PLC exchange markets under exact and approximate equilibria. *STOC 2017 Proc. 49th Annu. ACM SIGACT Sympos. Theory Comput.* (Association for Computing Machinery, New York), 890–901.
- Ghiyasvand M, Orlin JB (2012) A simple approximation algorithm for computing Arrow-Debreu prices. *Oper. Res.* 60(5):1245–1248.
- Goel G, Vazirani V (2011) A perfect price discrimination market model with production, and a rational convex program for it. *Math. Oper. Res.* 36(4):762–782.
- Goldberg AV, Tarjan RE (1989) Finding minimum-cost circulations by canceling negative cycles. *J. ACM.* 36(4):873–886.
- Grötschel M, Lovász L, Schrijver A (1988) *Geometric Algorithms and Combinatorial Optimization, Algorithms and Combinatorics*, vol. 2 (Springer Verlag, Berlin).
- Hochbaum DS, Naor J (1994) Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM J. Comput.* 23(6):1179–1192.
- Jain K (2007) A polynomial time algorithm for computing an Arrow-Debreu market equilibrium for linear utilities. *SIAM J. Comput.* 37(1):303–318.
- Jain K, Mahdian M, Saberi A (2003) Approximating market equilibria. Arora S, Jansen K, Rolim JDP, Sahai A, eds. *RANDOM 2003 APPROX 2003 Approximation Randomization Combinatorial Optimization Algorithms Techniques, Lecture Notes in Computer Science*, vol. 2764 (Springer, Berlin), 98–108.
- Kamiyama N (2019) A note on balanced flows in equality networks. *Inform. Process. Lett.* 145:74–76.
- Lemke CE (1965) Bimatrix equilibrium points and mathematical programming. *Management Sci.* 11(7):681–689.
- Megiddo N (1983) Toward a genuinely polynomial algorithm for linear programming. *SIAM J. Comput.* 12(2):347–353.
- Nenakov EI, Primak ME (1983) One algorithm for finding solutions of the Arrow-Debreu model. *Kibernetika* 3:127–128.

- Olver N, Végh LA (2017) A simpler and faster strongly polynomial algorithm for generalized flow maximization. *STOC 2017 Proc. 49th Annu. ACM SIGACT Sympos. Theory Comput.* (Association for Computing Machinery, New York), 100–111.
- Orlin JB (1993) A faster strongly polynomial minimum cost flow algorithm. *Oper. Res.* 41(2):338–350.
- Orlin JB (2010) Improved algorithms for computing Fisher’s market clearing prices. *STOC’10 Proc. 42nd ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 291–300.
- Orlin JB (2013) Max flows in  $O(nm)$  time, or better. *STOC’13 Proc. 45th Annu. ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 765–774.
- Shmyrev VI (2009) An algorithm for finding equilibrium in the linear exchange model with fixed budgets. *J. Appl. Indust. Math.* 3(4):505–518.
- Tardos É (1985) A strongly polynomial minimum cost circulation algorithm. *Combinatorica* 5(3):247–255.
- Tardos É (1986) A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.* 34(2):250–256.
- Varian H (1974) Equity, envy and efficiency. *J. Econom. Theory* 29(2): 217–244.
- Vavasis SA, Ye Y (1996) A primal-dual interior point method whose running time depends only on the constraint matrix. *Math. Program.* 74(1):79–120.
- Vazirani V (2010) Spending constraint utilities with applications to the adwords market. *Math. Oper. Res.* 35(2):458–478.
- Vazirani VV (2012) The notion of a rational convex program, and an algorithm for the Arrow-Debreu Nash bargaining game. *J. ACM* 59(2):7.
- Végh LA (2013) Concave generalized flows with applications to market equilibria. *Math. Oper. Res.* 39(2):573–596.
- Végh LA (2016) A strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. *SIAM J. Comput.* 45(5):1729–1761.
- Végh LA (2017) A strongly polynomial algorithm for generalized flow maximization. *Math. Oper. Res.* 42(2):179–211.
- Walras L (1874) *Eléments d’Economie Politique Pure, ou Théorie de la Richesse Sociale* (in French). [English translation: *Elements of Pure Economics; or, the Theory of Social Wealth*] (American Economic Association and the Royal Economic Society, Corbaz & Cie, Lausanne).
- Ye Y (2005) A new complexity result on solving the Markov decision problem. *Math. Oper. Res.* 30(3):733–749.
- Ye Y (2008) A path to the Arrow–Debreu competitive market equilibrium. *Math. Program.* 111(1–2):315–348.
- Ye Y (2011) The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. *Math. Oper. Res.* 36(4):593–603.

---

**Jugal Garg** is an assistant professor in the Department of Industrial and Enterprise Systems Engineering at the University of Illinois at Urbana-Champaign. He is broadly interested in computational aspects of economics and game theory, design and analysis of algorithms, and mathematical programming.

**László A. Végh** is a professor of mathematics at the London School of Economics and Political Science. He is broadly interested in fundamental questions in algorithms and optimisation: exact and approximation algorithms for problems related to network design, flows, matchings, and equilibrium computation, with a particular focus on strongly polynomial computability.