Combinatorial Algorithms for General Linear Arrow-Debreu Markets

Bhaskar Ray Chaudhury
MPI for Informatics, Saarland Informatics Campus, Graduate School of Computer Science Saarbrücken, Germany
braycha@mpi-inf.mpg.de

Kurt Mehlhorn
MPI for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
mehlhorn@mpi-inf.mpg.de

Abstract

We present a combinatorial algorithm for determining the market clearing prices of a general linear Arrow-Debreu market, where every agent can own multiple goods. The existing combinatorial algorithms for linear Arrow-Debreu markets consider the case where each agent can own all of one good only. We present an $\tilde{\mathcal{O}}((n+m)^7 \log^3(UW))$ algorithm where n, m, U and W refer to the number of agents, the number of goods, the maximal integral utility and the maximum quantity of any good in the market respectively. The algorithm refines the iterative algorithm of Duan, Garg and Mehlhorn using several new ideas. We also identify the hard instances for existing combinatorial algorithms for linear Arrow-Debreu markets. In particular we find instances where the ratio of the maximum to the minimum equilibrium price of a good is $U^{\Omega(n)}$ and the number of iterations required by the existing iterative combinatorial algorithms of Duan, and Mehlhorn and Duan, Garg, and Mehlhorn are high. Our instances also separate the two algorithms.

2012 ACM Subject Classification Theory of computation → Mathematical optimization

Keywords and phrases Linear Exchange Markets, Equilibrium, Combinatorial Algorithms

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2018.26

Related Version A full version of the paper is available at [2], `https://arxiv.org/abs/1810.01237`.

1 Introduction

In a linear Arrow-Debreu market, there is a set B of n agents and a set G of m divisible goods. We will refer to an individual agent by b_i for $i \in [n]$ and to an individual good by g_j for $j \in [m]$. Each agent comes with a basket of goods to the market, more precisely, agent b_i owns $w_{ij} \geq 0$ units of good g_j. The total supply of g_j is then $\sum_i w_{ij}$ units. Moreover, the agents have utilities over the goods and $u_{ij} \geq 0$ is the utility derived by agent b_i from one unit of good g_j.

The goal is to find a positive price vector $p \in \mathbb{R}^m_{\geq 0}$ and a non-zero flow $f \in \mathbb{R}^{n \times m}_{\geq 0}$ such that:

a) For all $j \in [m]$: $\sum_i f_{ij} = \sum_i w_{ij} p_j$ (all goods are completely sold)
b) For all $i \in [n]$: $\sum_j w_{ij} p_j = \sum_j f_{ij}$ (agents spend all their income)
c) $f_{ij} > 0$ implies $u_{ij}/p_j = \max_{\ell \in [m]} u_{i\ell}/p_\ell$ (only bang-for-buck spending)

38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018).
Editors: Sumit Ganguly and Paritosh Pandya; Article No. 26; pp. 26:1–26:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Such a price vector is called a vector of equilibrium prices. In a), the left hand side is the total flow (of money) into g_j and the right hand side is the total value of all units of good g_j. In b), the left hand side is the income of agent b_i under prices p and the right hand side is his spending. In c), $\max_{\ell\in[m]} u_{i\ell}/p_\ell$ is the maximum ratio of utility to price (bang-for-buck) that agent b_i can achieve. Agents spend only money on goods that give them the maximum bang-for-buck. For agent b_i and good g_j, we use $x_{ij} = f_{ij}/p_j$ for the amount of g_j allocated to agent b_i. A price vector p and an flow f as above is called a *market equilibrium*.

We make the standard assumption that every agent likes at least one good, i.e., for all i, $\max_j u_{ij} > 0$, and that each good is liked by some agent, i.e., for all j, $\max_i u_{ij} > 0$. We also make the nonstandard assumption that for every proper subset $B' \subset B$ there is at least one good g_j that is not completely owned by the agents in B' and such that at least some $b_i \in B'$ is interested in g_j, i.e., $w_{kj} > 0$ for some $b_k \notin B'$ and $u_{ij} > 0$ for some $b_i \in B'$. In other words, there is no subset of agents that are only interested in the goods completely owned by them. References [15, 7, 11] show how to remove the nonstandard assumption. We assume that utilities u_{ij} and weights w_{ij} are integral and use $U = \max_{i\in[n],j\in[m]} u_{ij}$ to denote the maximum utility and $W = \max_{j\in[m]} \max_{i\in[n]} w_{ij}$ to denote the maximum weight. Then the budget available to any agent is bounded by $\sum_j w_{ij}p_j \le nW \max_j p_j$.

Linear Exchange markets were introduced by Walras [19] back in 1874. Walras also argued that equilibrium prices exist. The first rigorous proof for the existence of equilibirum under strong assumptions was given by Wald [18]. Arrow and Debreu [1] gave the proof for the existence of equilibrium when the utility functions are concave. There has been substantial algorithmic research put into determining the equilibrium prices since the 60s. Codenotti et al. [3] gives a surveys the algorithmic literature before 2004. While there are strongly polynomial approximation schemes for determining the equilibrium prices [16, 14, 9], the existence of a strongly polynomial exact algorithms still remains as an open question. There have been exact finite algorithms [12, 13], exact weakly polynomial time algorithms [15, 20, 11, 10] and the characterization of the equilibrium prices as a solution set of a convex program [7, 17].

A market equilibrium can be found in time polynomial in n, m, $\log U$ and $\log W$ by a number of different algorithms. Jain [15] and Ye [20] gave algorithms based on the ellipsoid and the interior point method, respectively, and Duan and Mehlhorn [11] and Duan, Garg, and Mehlhorn [10] described combinatorial algorithms. The algorithm by Ye has a running time of $\mathcal{O}(\max(n, m)^8(\log UW)^2)$, see [10, footnote on page 2].

The combinatorial algorithms actually only solve a special case: $m = n$ and each agent is the sole owner of a good, i.e., $w_{ii} = 1$, and $w_{ij} = 0$ for $i \ne j$. The algorithm in [10] solves the special case in time $\mathcal{O}(n^7 \log^3(nU))$. A reduction for reducing the general case to the special case is known, see Section 2. However, it turns a general problem with n agents and m goods into a special problem with nm goods and hence leads to a running time of $\tilde{\mathcal{O}}((nm)^7\text{poly}(\log(U)))$ (ignoring poly logarithmic dependencies on n). In an unpublished note, Darwish and Mehlhorn [6, 4] have shown how to extend the algorithm in [11] to the general problem without going through the reduction. The resulting running time is $\mathcal{O}(\max(n, m)^{10} \log^2(\max(n, m)UW))$. They were unable to generalize the approach in [10].

Our Contribution. Our contribution is twofold: a combinatorial algorithm for the general problem and examples that are difficult for the algorithms in [11, 10]. We give a combinatorial algorithm for the general problem with running time $\mathcal{O}((n + m)^7 \log^3(nmUW))$. The algorithm refines the algorithm in [10] by several new ideas. We discuss them in Section 2. In particular, in [10], the number of iterations compared to [11] is reduced by a factor of

$\Omega(n)$ by a modified price update rule . The modified update rule is subtle and heavily relies on the fact that there is a one to one correspondence between an agent and a good (one agent owns all of one good only) and this is not true in the general scenario and several of their crucial arguments break down. We come up with a novel price update rule that also highlights some new structure in the problem.

We also give examples that are difficult for the algorithms in [11, 10] amd where the equilibrium prices are exponential in U. Both algorithms are iterative and need $\mathcal{O}(n^5 \log(U))$ and $\mathcal{O}(n^4 \log(nU))$ iterations respectively. The examples force the algorithms into $\tilde{\Omega}(n^{4+\frac{1}{3}} \log(U))$ and $\tilde{\Omega}(n^4 \log(U))$ iterations respectively (ignoring poly logarithmic dependencies on n). They separate the two algorithms.

2 Determining the equilibrium price vector of the general linear Arrow-Debreu market

For completeness we first give the reduction from the general case to the special case (where each agent owns all of one good only). This reduction is well-known. For each positive w_{ij} we create an agent b_{ij} and a good g_{ij} owned by this agent. There is one unit of good g_{ij}. If $w_{ij} = 0$, there is no good g_{ij} and no agent b_{ij}. We interpret g_{ij} as the goods g_j owned by b_i and b_{ij} as a copy of agent b_i. We define the utility derived by the agent b_{ij} from one unit of good $g_{\ell k}$ as $\tilde{u}_{ij,\ell k} = w_{\ell k} \cdot u_{ik}$; here the factor u_{ik} reflects that b_{ij} is a copy of agent b_i and $g_{\ell k}$ is a copy of g_k and the factor $w_{\ell k}$ reflects that b_ℓ owns $w_{\ell k}$ units of good g_k but there is only copy of good $g_{\ell k}$.

▶ **Lemma 1.** *Let p and f be the market clearing price vector and the corresponding money flow for the above instance of the special case with nm agents and goods. Then $\frac{p_{\ell k}}{w_{\ell k}}$ does not depend on ℓ, but only on k. Let $\hat{p}_k = \frac{p_{\ell k}}{w_{\ell k}}$ and $\hat{f}_{ik} = \sum_{j \in [m]} \sum_{\ell \in [n]} f_{ij,\ell k}$. Then $\hat{p}$ and $\hat{f}$ are the market clearing price vector and corresponding money flow for general case with utility matrix u and weight matrix w.*

Proof. Assume $\frac{w_{\ell k}}{p_{\ell k}} > \frac{w_{hk}}{p_{hk}}$. Let b_{ij} be any arbitrary agent. Then

$$\frac{\tilde{u}_{ij,\ell k}}{p_{\ell k}} = \frac{w_{\ell k} \cdot u_{ik}}{p_{\ell k}} > \frac{w_{hk} \cdot u_{ik}}{p_{hk}} = \frac{\tilde{u}_{ij,hk}}{p_{hk}}.$$

Hence agent b_{ij} prefers good $g_{\ell k}$ over good g_{hk}. Since b_{ij} is arbitrary, g_{hk} will not be sold at all, a contradiction. Thus $\frac{w_{\ell k}}{p_{\ell k}} = \frac{w_{hk}}{p_{hk}}$ for all ℓ and h. Now, it is easy to verify that every agent invests in goods that give him maximum utility to price ratio. Assume $\hat{f}_{ik} > 0$. Then for any arbitrary k' and ℓ', there is a j and ℓ, such that

$$\frac{u_{ik}}{\hat{p}_k} = \frac{\tilde{u}_{ij,\ell k}}{w_{\ell k} \cdot \hat{p}_k} = \frac{\tilde{u}_{ij,\ell k}}{p_{\ell k}} \geq \frac{\tilde{u}_{ij,\ell' k'}}{p_{\ell' k'}} = \frac{\tilde{u}_{ij,\ell' k'}}{w_{\ell' k'} \cdot \hat{p}_{k'}} = \frac{u_{ik'}}{\hat{p}_{k'}}.$$

The equilibrium flow constraints are also easily verifiable,

$$\sum_{i \in [n]} \hat{f}_{ik} = \sum_{i \in [n]} \sum_{j \in [m]} \sum_{\ell \in [n]} f_{ij,\ell k} = \sum_{\ell \in [n]} p_{\ell k} = \sum_{\ell \in [n]} w_{\ell k} \cdot \hat{p}_k$$

$$\sum_{k \in [m]} \hat{f}_{ik} = \sum_{k \in [m]} \sum_{j \in [m]} \sum_{\ell \in [n]} f_{ij,\ell k} = \sum_{j \in [m]} p_{ij} = \sum_{j \in [m]} w_{ij} \cdot \hat{p}_j$$ ◀

We now give our algorithm that does not rely on this reduction.

Algorithm 1 Combinatorial algorithm for determining the equilibirum prices in the general linear Arrow-Debreu market.

Set $p_i \leftarrow 1 \quad \forall j \in [n]$.
Set $\varepsilon \leftarrow 1/(8 \cdot (n+m)^{4(n+m)}(UW)^{3(n+m)})$.
while $\|r_f\|_2 > \varepsilon$, where f is a balanced flow in N_p **do**
    Let S be the set of high-surplus agents w.r.t f in N_p.
    $x \leftarrow \min(x_{\text{eq}}, x_{23}, x_{24}, x_{13}, x_2, x_{\max})$.
    Multiply prices of goods in $\Gamma(S)$ by x and update f, p to f' and p' as in (1) - (2) and N_p to $N_{p'}$.
    Let f'' be the balanced flow in $N_{p'}$.
    Set $p \leftarrow p'$ and $f \leftarrow f''$.
end while
Round p to equilibrium prices.

2.1 The Algorithm

Algorithm 1 shows the algorithm. Similar to the algorithms in [8, 11, 10], the algorithm is iterative and flow based. For the description of the algorithm, we need the concepts of an equality network, of a balanced flow, of the set of high-surplus buyers, and of the flow- and price-update.

Equality Network N_p. For a price vector p the equality network N_p is a flow network with vertices $s \cup t \cup B \cup G$ and edges

- (s, b_i) with capacity $\sum_{j\in[m]} w_{ij} \cdot p_j$ for all $i \in [n]$.
- (g_j, t) with capacity $\sum_{i\in[n]} w_{ij} \cdot p_j$ for all $j \in [m]$.
- (b_i, g_j) with capacity ∞ iff $u_{ij}/p_j \geq u_{ik}/p_k$ for all $k \in [m]$.

For any $B' \subseteq B$ let $\Gamma(B') = \{g_j \mid (b_i, g_j) \in N_p \text{ for some } b_i \in B'\}$ denote the neighborhood of B' in N_p (all the goods agents in B' may invest on).

Surpluses and Surplus vector r_f. Let f be a valid flow in the equality network N_p, and let f_{ij}, f_{si} and f_{jt} denote the flow along the edge (b_i, g_j), (s, b_i) and (g_j, t) respectively. We define the surplus $r_f(b_i)$ of the agent b_i and $r_f(g_j)$ of the good g_j, as $r_f(b_i) = \sum_{j\in[m]} w_{ij} \cdot p_j - f_{si}$ and $r_f(g_j) = \sum_{i\in[n]} w_{ij} \cdot p_j - f_{jt}$ respectively. Surpluses are always non-negative. We define the surplus vector $r_f \in \mathbb{R}^n$ as $\langle r_f(b_1), r_f(b_2), \ldots r_f(b_n)\rangle$.

Balanced Flow: A balanced flow f is a valid flow in N_p with minimum norm $\|r_f\|_2^2$. Every balanced flow is a maximum flow. Additionally, if f is a balanced flow and f_{ij} and $f_{i'j}$ are positive, then the surpluses of the agents b_i and $b_{i'}$ are the same and if (b_i, g_j) and $(b_{i'}, g_j)$ are edges of N_p and $r_f(b_i) > r_f(b_{i'})$ then $f_{i'j} = 0$. This essentially follows from the fact that the L2 norm of a vector reduces as the components move closer to each other in magnitude (while the L1 norm remain constant). The following algorithmic property of balanced flows will be useful.

▶ **Lemma 2.** *Balanced Flows can be computed with at most n max flow computations [8] and by one parameterized flow computation [5].*

High Surplus Agents S and Goods in High Demand $\Gamma(S)$. Let f be a balanced flow and assume that there is surplus. The high surplus edges and high demand goods w.r.t. f in

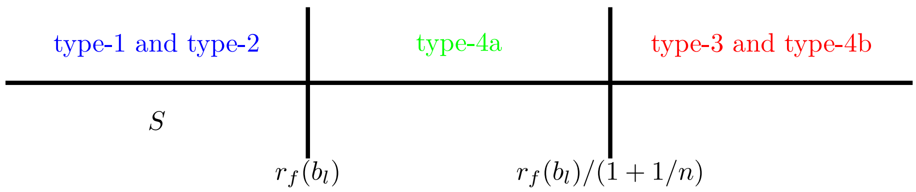

Figure 1 The horizontal line indicates the agents ordered by decreasing surplus from left to right. b_ℓ is the agent of smallest surplus in S. All agents with surplus in $r_f(b_\ell), r_f(b_\ell)/(1+1/n)]$ are type-4a agents. They have no outflow and do not own goods in $\Gamma(S)$.

N_p are defined exactly as in [10]. Renumber the agents in order of decreasing surplus so that b_1 has the highest surplus and b_n has the lowest surplus. Let ℓ be minimal such that $r_f(b_\ell) > r_f(b_{\ell+1})$ and for every k such that $r_f(b_\ell) > r_f(b_k) \geq r_f(b_\ell)/(1+1/n)$, $f_{sk} = 0$ and $w_{kj} = 0$ for every $g_j \in \Gamma(S)$, where $S = \{b_1, \ldots, b_\ell\}$. If no such ℓ exists, let $\ell = n$. We refer to S as the set of high surplus agents and to $\Gamma(S)$ as the set of high demand goods. The surplus of the goods in $\Gamma(S)$ is zero since the agents in S have positive surplus. There is no flow on edges $(b_i, g_j) \in N_p$ with $b_i \notin S$ and $g_j \in \Gamma(S)$.

The algorithm in [10] that determines S in time $\mathcal{O}(n^2)$ can be easily generalized and we do not discuss it here.

Price and Flow Update. Like the earlier combinatorial algorithms, our algorithm is a multiplicative price update algorithm. It works in phases and in each phase we compute the balanced flow f in N_p and determine the high surplus agents S and high demand goods $\Gamma(S)$. We increase the prices of the goods in $\Gamma(S)$ as well as the money flow into them by the same factor $x > 1$ (so we change the flow f in N_p to f' in $N_{p'}$). Formally,

$$f'_{ij} = \begin{cases} x \cdot f_{ij} & g_j \in \Gamma(S) \\ f_{ij} & g_j \notin \Gamma(S) \end{cases} \qquad f'_{si} = \begin{cases} x \cdot f_{si} & b_i \in S \\ f_{si} & b_i \notin S \end{cases} \tag{1}$$

$$f'_{jt} = \begin{cases} x \cdot f_{jt} & g_j \in \Gamma(S) \\ f_{jt} & g_j \notin \Gamma(S) \end{cases} \qquad p'_i = \begin{cases} x \cdot p_i & g_j \in \Gamma(S) \\ p_i & g_j \notin \Gamma(S) \end{cases} \tag{2}$$

The Factor x. We define $x = \min(x_{\text{eq}}, x_{23}, x_{24}, x_{13}, x_2, x_{\max})$, where the quantities on the right are defined below.

Since we increase the prices of all the goods in $\Gamma(S)$ by the same factor x, the only equality edges that may disappear are the ones that connect an agent from $B \setminus S$ to a good from $\Gamma(S)$. Since f is a balanced flow and the agents in S have strictly higher surplus than the ones in $B \setminus S$, the edges from agents in $B \setminus S$ to goods in $\Gamma(S)$ in N_p carry no flow and hence them disappearing will not lead to a violation of the flow constraints. The new edges that appear will connect an agent in S to a good in $G \setminus \Gamma(S)$. Therefore we define,

$$x_{\text{eq}} = \min \left\{ \frac{u_{ij}}{p_j} \cdot \frac{p_k}{u_{ik}} \;\middle|\; b_i \in S, (b_i, g_j) \in N_p, g_k \notin \Gamma(S) \right\}$$

This is the minimum x at which a new equality edge appears in the network.

Next we consider how the surpluses of the agents are affected. Observe that $r_{f'}(b_i) = r_f(b_i) + x \cdot (\sum_{g_j \in \Gamma(S)} (w_{ij} p_j - f_{ij}))$. Therefore all the surpluses vary linearly with x. We now, introduce 5 classes of agents similar to the ones in [10]. Figure 1 illustrates this definition.

- Type-1 agent: An agent is a type-1 agent if it belongs to S and its surplus increases by the price change. Formally, b_i is type-1 if $b_i \in S$ and $\sum_{g_j \in \Gamma(S)} w_{ij} p_j > \sum_{g_j \in \Gamma(S)} f_{ij}$.

- Type-2 agent: An agent is a type-2 agent if it belongs to S and its surplus does not increase by the price change. Formally, b_i is type-2 if $b_i \in S$ and $\sum_{g_j \in \Gamma(S)} w_{ij} p_j \leq \sum_{g_j \in \Gamma(S)} f_{ij}$.
- Type-3 agent: An agent is type-3 agent if it does not belong to S and its surplus increases by the price change, i.e., it partially owns a good in $\Gamma(S)$. Formally, $b_i \notin S$ and $w_{ij} > 0$ for some $g_j \in \Gamma(S)$.
- Type-4a agent: An agent is a type-4a agent if it does not belong to S and has no ownership in a good in $\Gamma(S)$ and its surplus is at least $r_{\min}/(1 + 1/n)$, where $r_{\min}$ is the minimum surplus of any agent in S. Formally, $b_i \notin S$, $w_{ij} = 0$ for all $j \in \Gamma(S)$ and $r_f(b_i) \geq r_{\min}/(1 + 1/n)$. By definition of S such b_i have no outflow, i.e., $f_{si} = 0$, and no ownership of any good in $\Gamma(S)$, i.e., $w_{ij} = 0$ for any $j \in \Gamma(S)$.
- Type-4b agent: An agent is a type-4b agent if it does not belong to S and has no ownership in a good in $\Gamma(S)$ and its surplus is strictly less than $r_{\min}/(1 + 1/n)$, where $r_{\min}$ is the minimum surplus of any agent in S. Formally, $b_i \notin S$, $w_{ij} = 0$ for all $j \in \Gamma(S)$ and $r_f(b_i) < r_{\min}/(1 + 1/n)$.

We abbreviate the above sets of agents of each type as T_1, T_2, T_3, T_{4a} and T_{4b}. Notice that the surpluses of agents in $T_{4a} \cup T_{4b}$ remain unaffected by increase of price of goods in $\Gamma(S)$. We now define x_{23}, x_{24}, x_{13} as the minimal x such that $r_{f'}(b_i) = r_{f'}(b_j)$ for some $b_i \in T_2, b_j \in T_3$ and $b_i \in T_2, b_j \in T_{4b}$ and $b_i \in T_1, b_j \in T_3$ respectively. We also define x_2 as the minimal x when $r_{f'}(b_i) = 0$ where $b_i \in T_2$.

Finally,

$$x_{\max} = \begin{cases} 1 + \frac{1}{Rn^3} & \text{when } \min_{g_j \in \Gamma(S)} p_j < Rn^4 mW \\ 1 + \frac{1}{Rkn^2} & \text{when } \min_{g_j \in \Gamma(S)} p_j \geq Rn^4 mW, \end{cases}$$

where k is the number of agents in S that partially own a good in $\Gamma(S)$ and $R = 8e^2$. Note that k is at least the number of type-1 agents.[1] We distinguish light and heavy iterations. An iteration is light, if $p_j < Rn^4 mW$ for some good $g_j \in \Gamma(S)$ (some good in demand is not heavily priced), and heavy if $p_j \geq Rn^4 mW$ for all goods $g_j \in \Gamma(S)$ (all goods in demand are heavily priced).

Effect of an Iteration. We multiply the prices of the goods in $\Gamma(S)$ by x and update f, p to f' and p' as in (1)-(4) and N_p to $N_{p'}$. Note that the goods completely sold w.r.t. f in N_p are also completely sold with respect to $N_{p'}$. Maximizing and balancing the flow will not increase the surplus of a good from zero to a positive value. Thus all goods that are completely w.r.t. f in N_p are also completely sold w.r.t. the balanced flow f'' in $N_{p'}$. The 2-norm of $r_{f''}$ is at most the $L2$ norm of $r_{f'}$.

We stop updating the prices once the $L2$ norm of the surplus vector is at most $\varepsilon = 1/(8(n+m)^{4(n+m)}(U \cdot W)^{3(n+m)})$.

---

[1] In [10], $x_{\max}$ is defined as

$$x_{max} = \begin{cases} 1 + \frac{1}{Rn^3} & \text{if there are type-3 agents} \\ 1 + \frac{1}{Rkn^2} & \text{otherwise, where } k = \text{number of type-1 agents.} \end{cases}$$

The revised definition we change the classification of iterations. We do not classify them based on the type of the agents in S, but by looking at the smallest price of a good in $\Gamma(S)$ and we have a larger k (k is at least the number of type-1 agents).

2.2 Analysis of the Algorithm

Before we present the analysis of our algorithm we briefly indicate why the price-update scheme used in [10] does not generalize. The algorithm in [10] reduces the number of iterations of the algorithm in [11] by a multiplicative factor of $\Omega(n)$ by performing a more careful selection of the set S and crucial change in the price update rule. In particular they set

$$x_{max} = \begin{cases} 1 + \frac{1}{Rn^3} & \text{if there are type-3 agents} \\ 1 + \frac{1}{Rkn^2} & \text{otherwise, where } k = \text{number of type-1 agents.} \end{cases}$$

Since in the special case, every agent owns only all of one good, k equals the number of type-1 agents in S and the number of goods in $\Gamma(S)$ in all iterations that do not involve a type-3 agent. We enlist two crucial arguments that are necessary (not sufficient) to reduce the number of iterations by $\Omega(n)$ from that in [11] and their dependencies on k as follows,

1. There is a decrease in the $L2$ norm of the surplus vector in every balancing iteration without a type-3 agent. This argument crucially relies on k being equal to the number of type-1 agents.
2. The total multiplicative increase of the $L2$ norm of the surplus vector in all x_{max} iterations without a type-3 agent is at most $(nU)^{\mathcal{O}(n)}$. This claim relies on the fact that k equals the number of goods in $\Gamma(S)$.

In the general scenario we do not have such one to one correspondence between the agents and the goods. Therefore if we choose k to be either the number of type-1 agents in S or the number of goods in $\Gamma(S)$, then one of the claims from above will fail. Thus we need at least a different classification of the iterations or a different choice of k than the ones used in [10]. In our algorithm we do both and thereby highlighten more hidden structure in the problem.

Like the algorithms in [11, 10], in every iteration Algorithm 1 only increases the prices of goods that are completely sold (since f is a balanced flow and the agents in S have positive surplus, the goods in $\Gamma(S)$ will have zero surplus). Since the sum of surpluses of the agents equals the sum of surpluses of the goods, both are therefore non-decreasing during a price update. Also the goods completely sold w.r.t. f in N_p also remain sold w.r.t. f'' in $N_{p'}$ in every iteration. Therefore the sum of surpluses of the agents is non-increasing throughout the algorithm. Initially (when the price of every good g_j is 1) this sum is at most nmW.

▶ **Observation 3.** *The $L1$ norm of the surplus vector is non-increasing throughout the algorithm and is at most nmW.*

2.2.1 An Upper Bound on the Maximum Price

Observe that in every iteration of the algorithm there is a good with unit price. This follows from the fact that the goods that are completely sold stay completely sold and since only the prices of goods that are completely sold are increased, the price of goods that are not completely sold is equal to the initial price and hence equal to one. We now derive an upper bound on the maximum price of a good.

▶ **Lemma 4.** *At any time during the course of the algorithm the maximum price is at most* $\max(2, U)^{m-1} \cdot W^{2m-2}$ *and the maximum budget is at most* $\max(2, U)^m \cdot W^{2m}$.

Proof. Let us renumber the goods in increasing order of their prices. We show that $p_i \leq \max(2, U)^{i-1} \cdot W^{2i-2}$ by induction on i. The smallest price is 1 and this establishes the induction base. Consider an arbitrary i and let $\hat{G}$ denote the set of goods $\{g_{i+1}, ..., g_m\}$ and $\hat{B}$ be the agents investing on the goods in $\hat{G}$. We will derive a bound on p_{i+1}. We distinguish two cases.

Assume first that there is an agent in $b_h \in \hat{B}$ that is also interested in a good in $G \setminus \hat{G}$; say b_h invests on good $g_j \in \hat{G}$ and is interested in good $g_\ell \in G \setminus \hat{G}$. Then $u_{hj}/p_j \geq u_{h\ell}/p_\ell$ and hence $p_{i+1} \leq p_j \leq U \cdot p_\ell \leq U \cdot p_i \leq U \cdot \max(2,U)^{i-1} \cdot W^{2i-2} \leq \max(2,U)^i \cdot W^{2i}$.

Assume next that the agents in $\hat{B}$ are only interested in the goods in $\hat{G}$. Then there must be a good in $\hat{G}$, say g_k, that is partially owned by an agent in $B \setminus \hat{B}$. Otherwise, the agents in $\hat{B}$ will only be interested in goods completely owned by them. Therefore let $b_h \in B \setminus \hat{B}$ be the agent that partially owns g_k. The budget m_h of b_h is at least $w_{hk}p_k$ and hence at least p_k. Since b_h invests only in goods in $G \setminus \hat{G}$, its budget is at most the total value of the goods in $G \setminus \hat{G}$. Thus

$$m_h \leq W \cdot \sum_{j \in [i]} p_j \leq W \cdot \sum_{j \in [i]} \left(\max(2,U)^{j-1} W^{2j-2}\right) \leq W \max(2,U)^i W^{2i-1} = \max(2,U)^i W^{2i}$$

and hence $p_{i+1} \leq p_k \leq \max(2,U)^i W^{2i}$. The maximum budget of an agent is at most W times the total price of all goods and hence is bounded by $W \cdot \sum_{j \in [m]} \left(\max(2,U)^{j-1} \cdot W^{2j-2}\right) \leq \max(2,U)^m \cdot W^{2m}$. ◀

In particular, when $n = m$ and w is an identity matrix, we have an upper bound of $\max(2,U)^{n-1}$ for the highest price of a good in contrast to $\max(n,U)^n$ in [11, 10]. Also note that the maximum price of a good and the maximum budget of an agent is independent of the number of agents. We now separately bound the $x_{\max}$-iterations ($x = x_{\max}$) and balancing iterations ($x < x_{\max}$).

2.2.2 Bounding the number of $x_{\max}$-iterations

In this section we will be bounding the light and the heavy $x_{\max}$-iterations separately. For bounding both classes of iterations, we use upper bounds on the prices of the goods. A more aggressive price update scheme is used for the heavy $x_{\max}$-iterations as the prices of all goods in $\Gamma(S)$ in such iterations are high. Such aggressive price update may apparently result in a significant multiplicative increase in the $L2$ norm of the surplus vector. We address this concern in the next subsection. We first show that $x_{\max}$-iterations where there is at least one type-3 agent are light.

▶ **Lemma 5.** *$x_{\max}$-iterations with at least one type-3 agent are light.*

Proof. Let b_i be a type-3 agent. Then there is a $g_j \in \Gamma(S)$ with $w_{ij} > 0$. The additive increase in the surplus of a type-3 agent b_i during an $x_{\max}$-iteration is at least $w_{ij} \cdot p_j / Rn^3$. Since the total surplus is always at most nmW (by Observation 3), the increase in the surplus of any agent is at most nmW. This immediately implies that $p_j \leq Rn^4 mW$. ◀

Now we bound the number of light $x_{\max}$-iteration.

▶ **Lemma 6.** *The number of light $x_{\max}$-iterations is at most $10R^2n^3m\log(nmW)$.*

Proof. Assume otherwise. Then there exists a good g_j that is the minimum priced good in $\Gamma(S)$ in more than $10R^2n^3\log(nmW)$ iterations. Before the last such iteration, $p_j > (1+\frac{1}{Rn^3})^{10R^2n^3\log(nmW)} > e^{5R\log(nmW)} = n^{5R}m^{5R}W^{5R} > Rn^4mW$, which is a contradiction. The second to last inequality uses the fact that $1 + x > e^{\frac{x}{2}}$ for $0 < x \leq 1$. ◀

We turn to heavy $x_{\max}$-iterations. For such iterations, there exists no type-3 agent and hence the goods in $\Gamma(S)$ are completely owned by the agents in $T_1 \cup T_2$. Thus

$$\sum_{b_i \in T_1 \cup T_2} \sum_{g_j \in \Gamma(S)} f_{ij} = \sum_{g_j \in \Gamma(S)} \sum_{i \in T_1 \cup T_2} w_{ij}p_j = \sum_{b_i \in T_1} \sum_{g_j \in \Gamma(S)} w_{ij} \cdot p_j + \sum_{b_i \in T_2} \sum_{g_j \in \Gamma(S)} w_{ij} \cdot p_j,$$

and hence

$$\sum_{b_i \in T_1} \sum_{g_j \in \Gamma(S)} (w_{ij} \cdot p_j - f_{ij}) = \sum_{b_i \in T_2} \sum_{g_j \in \Gamma(S)} (f_{ij} - w_{ij} \cdot p_j). \tag{3}$$

Note that type-1 and type-2 agents can own goods outside of $\Gamma(S)$. However the above relation will help us prove that this "excess budget" is at most nmW. In fact the following lemma plays a pivotal role to bound the multiplicative increase in the $L2$ norm of the surplus vector in the next subsubsection.

▶ **Lemma 7.** *For every agent $b_i \in S$ in a heavy $x_{\max}$-iteration, $\sum_{g_j \notin \Gamma(S)} w_{ij} p_j \leq nmW$.*

Proof. For $b_i \in S$, we have $r_f(b_i) = \sum_{g_j \notin \Gamma(S)} w_{ij} p_j + \sum_{g_j \in \Gamma(S)} (w_{ij} p_j - f_{ij})$. For $b_i \in T_1$, this implies $\sum_{g_j \notin \Gamma(S)} w_{ij} p_j \leq r_f(b_i) \leq nmW$. For $b_i \in T_2$, using (3)

$$\sum_{g_j \notin \Gamma(S)} w_{ij} p_j = r_f(b_i) + \sum_{g_j \in \Gamma(S)} (f_{ij} - w_{ij} p_j) \leq r_f(b_i) + \sum_{b_h \in T_2} \sum_{g_j \in \Gamma(S)} (f_{hj} - w_{hj} p_j)$$
$$= r_f(b_i) + \sum_{b_h \in T_1} \sum_{g_j \in \Gamma(S)} (w_{hj} p_j - f_{hj}) \leq r_f(b_i) + \sum_{b_h \in T_1} r_f(b_h) \leq nmW. \quad ◀$$

In a heavy $x_{\max}$-iteration, the price of any good in $\Gamma(S)$ is at least $Rn^4 mW$ and hence the budget of any agent that partially owns a good in $\Gamma(S)$ is at least that much. By the above, the ownership of the goods outside $\Gamma(S)$ contribute very little to the budget of such agents. Thus any multiplicative increment on the prices of the goods in $\Gamma(S)$ will inflict an almost equal multiplicative increase in the budget of such agents.

▶ **Lemma 8.** *The number of heavy $x_{\max}$-iterations is $\mathcal{O}(n^3 m \cdot \log(WU))$.*

Proof. Consider any heavy $x_{\max}$ iteration and let m_i denote the budget of agent b_i. For any agent b_i that partially owns a good in $\Gamma(S)$, $m_i \geq Rn^4 mW$. The budget m_i of any agent b_i that partially owns a good in $\Gamma(S)$, increases as follows (new budget denoted by m_i'):

$$\begin{aligned}
m_i' &= \sum_{g_j \notin \Gamma(S)} w_{ij} p_j + (1 + \frac{1}{Rkn^2}) \cdot \sum_{g_j \in \Gamma(S)} w_{ij} p_j \\
&\geq (1 + \frac{1}{Rkn^2}) \cdot (1 + \frac{1}{n^3})^{-1} \cdot (1 + \frac{1}{n^3}) \cdot \sum_{g_j \in \Gamma(S)} w_{ij} p_j \\
&= (1 + \frac{1}{Rkn^2}) \cdot (1 + \frac{1}{n^3})^{-1} \cdot \left(\sum_{g_j \in \Gamma(S)} w_{ij} p_j + \frac{\sum_{g_j \in \Gamma(S)} w_{ij} p_j}{n^3} \right) \\
&\geq (1 + \frac{1}{Rkn^2}) \cdot (1 + \frac{1}{n^3})^{-1} \cdot \left(\sum_{g_j \in \Gamma(S)} w_{ij} p_j + nmW \right) && \text{since } \sum_{g_j \in \Gamma(S)} w_{ij} p_j \geq Rn^4 mW \\
&\geq (1 + \frac{1}{Rkn^2}) \cdot (1 + \frac{1}{n^3})^{-1} \cdot \left(\sum_{g_j \in \Gamma(S)} w_{ij} p_j + \sum_{g_j \notin \Gamma(S)} w_{ij} p_j \right) && \text{since } \sum_{g_j \notin \Gamma(S)} w_{ij} p_j \leq nmW \\
&\geq (1 + \Omega(\frac{1}{kn^2})) \cdot m_i.
\end{aligned}$$

Let $M = \prod_{i \in [n]} m_i$. Since $m_i \leq (\max(2, U) W^2)^m$ we have $\log(M) \leq nm \log(UW)$ always. Also $\log M \geq 0$ initially. At any heavy $x_{\max}$ iteration, $\log(M)$ increases by a additive factor of $\log((1 + \Omega(\frac{1}{kn^2}))^k) \in \Omega(\frac{1}{n^2})$. Since $\log(M)$ is non-decreasing in every iteration of the algorithm (since the prices of the goods and the budgets of the agents only increase), the number of heavy $x_{\max}$-iterations is $\mathcal{O}(n^3 m \log(WU))$. ◀

We have now bounded the total number of $x_{\max}$-iterations by $\mathcal{O}(n^3 m \log(nmUW))$.

2.2.3 On the Increase in the $L2$-Norm of the Surplus Vector in the $x_{\max}$-Iterations

The $L2$ norm of the surplus vector is minimal for the balanced flow. We just look at the difference in the $L2$ norm of the surpluses with respect to the flows f in N_p and f' in $N_{p'}$ (Updated flow as in 1 - (2)) . Note that this difference in surplus is at least as large as the difference between the $L2$ norm of the surplus vector with respect to f in N_p and f'' in $N_{p'}$ (balanced flow in $N_{p'}$ in Algorithm 1). This suffices as we are upper bounding the difference in the $L2$ norm of the surplus vector in this section.

In a light $x_{\max}$-iteration, the $L2$ norm of the surplus vector increases at most by a factor of $(1+\mathcal{O}(\frac{1}{n^3}))$ and thus the total multiplicative increase in the $L2$ norm of the surplus vector resulting from such iterations is $(1+\mathcal{O}(\frac{1}{n^3}))^{\mathcal{O}(n^3 m \cdot \log(nmW))} = (nmW)^{\mathcal{O}(m)}$.

We now bound the multiplicative increase resulting from heavy $x_{\max}$-iterations. Despite the more aggressive price update scheme in heavy $x_{\max}$-iterations, we can assure the same multiplicative increase. As in [10] we wish to prove that the ratio of the highest to the lowest surplus of the agents in S is at most $1+\mathcal{O}(\frac{k}{n})$). One possible approach is to show that the number of distinct surpluses in S is $\mathcal{O}(k)$ (in that case the ratio will be $(1+\frac{1}{n})^{\mathcal{O}(k)} = 1+\mathcal{O}(\frac{k}{n})$). In [10], this is relatively easy to argue, as goods and agents are in one-to-one correspondence and all agents having positive outflow to a good g have same surplus (by the property of balanced flow). This immediately implies that there are at most $2k$ distinct surpluses of the agents in S (additional k for agents with zero outflow that own one of the goods in $\Gamma(S)$). This argument does not hold in the general scenario as the number of goods can be much larger $|S|$. However Lemma 7 gives us a useful structure in the equality network.

▶ **Lemma 9.** *The total multiplicative increase resulting in the $L2$ norm of the surplus vector in heavy $x_{\max}$-iterations is $(WU)^{\mathcal{O}(m)}$.*

Proof. Let S' be the set of agents in S that partially own some good in $\Gamma(S)$ and $k = |S'|$. Let S'' be the set of agents in S with positive outflow. Any agent in S' has a budget of at least Rn^4mW (follows from the definition of heavy x_{max} iteration) and therefore has positive outflow (since its surplus is at most nmW by Observation 3). Thus $S' \subseteq S''$. By Lemma 7, the budget of any agent in $S \setminus S'$ is at most nmW and hence the total outflow from agents in $S \setminus S'$ is at most n^2mW. Therefore any good in $\Gamma(S)$ must have inflow from an agent in S' and hence the surplus of any agent in S'' is equal to the surplus of some agent in S'.

Let $r_1 > r_2 > \ldots > r_h$ with $h \leq k$ be the distinct surplus values of the agents in S''. Agents in $S \setminus S''$ have no outflow and no ownership of any good in $\Gamma(S)$. Therefore $r_{i+1} \geq r_i/(1+1/n)$ by definition of S for $1 \leq i < h$.

From $r_i \leq (1+\frac{1}{n})r_{i+1}$ for all i, we conclude $r_1 \leq (1+\frac{1}{n})^k r_h \leq (1+\frac{2k}{n})r_h$. Therefore for any type-1 agent b_i, we can claim that $r_f(b_i) < (1+\frac{2k}{n})r_h$. Let $r_{f'}$ be the surplus vector after the $x_{\max}$-iteration. Since there are no type-3 agents in this iteration, only the surpluses of the type-1 and type-2 agents belonging to S'' are affected (Agents belonging to $S \setminus S''$ have no ownership of goods in $\Gamma(S)$ and no outflow also, so their surpluses are unchanged when we change f to f') . Let δ_i denote the increase in the surplus of a type-1 agent b_i and let μ_j denote the decrease of surplus of a type-2 agent b_j. Note that $\sum_{b_i \in T_1} \delta_i = \sum_{b_i \in T_2} \mu_i \leq \frac{1}{Rkn^2} \sum_{b_i \in T_1} r_f(b_i)$. Then,

$$\begin{aligned}
\|r_{f'}\|_2^2 - \|r_f\|_2^2 &= \sum_{b_i \in T_1} ((r_f(b_i) + \delta_i)^2 - r_f(b_i)^2) - \sum_{b_i \in T_2} (r_f(b_i)^2 - (r_f(b_i) - \mu_i)^2) \\
&= 2 \sum_{b_i \in T_1} r_f(b_i)\delta_i - 2 \sum_{b_i \in T_2} r_f(b_i)\mu_i + \sum_{b_i \in T_1} \delta_i^2 + \sum_{b_i \in T_2} \mu_i^2 \\
&\le 2r_1 \sum_{b_i \in T_1} \delta_i - 2r_h \sum_{b_i \in T_2} \mu_i + \sum_{b_i \in T_1} \delta_i^2 + \sum_{b_i \in T_2} \mu_i^2 \\
&\le 2(r_1 - r_h) \sum_{b_i \in T_1} \delta_i + 2 \left(\sum_{b_i \in T_1} \delta_i \right)^2 \\
&\le 2((1 + \frac{2k}{n})r_h - r_h) \sum_{b_i \in T_1} \frac{1}{Rkn^2} \cdot r_f(b_i) + 2 \frac{1}{R^2k^2n^4} \left(\sum_{b_i \in T_1} r_f(b_i) \right)^2 \\
&\le \frac{4}{Rn^3} \sum_{b_i \in T_1} r_f(b_i)^2 + n \cdot \frac{2}{R^2k^2n^4} \sum_{b_i \in T_1} r_f(b_i)^2,
\end{aligned}$$

Thus $\|r_{f'}\|_2^2 \in (1 + \mathcal{O}(\frac{1}{n^3}))\|r_f\|_2^2$. Therefore the multiplicative increase in any heavy $x_{\max}$-iterations is $1 + \mathcal{O}(\frac{1}{n^3})$. Thus the total multiplicative increase in the $L2$ norm of the surplus vector in all heavy $x_{\max}$ iterations is at most $(1 + \mathcal{O}(\frac{1}{n^3}))^{\mathcal{O}(n^3 m \log(WU))} \in \mathcal{O}(WU)^{\mathcal{O}(m)}$. ◀

Thus the total multiplicative increase in all $x_{\max}$-iterations is at most $(nmUW)^{\mathcal{O}(m)}$.

2.2.4 Balancing Iterations

In the balancing iterations $x < x_{\max}$. First we discuss the case when $x = \min(x_{23}, x_{24}, x_{13}, x_2)$. Since the $L1$ norm of the surplus vector is non-increasing during such an iteration (by Observation 3), the total decrease in the surplus of the type-2 agents is at least the total increase in the surpluses of the type-1 and type-3 agents. So now we quantify the decrease in the $L2$ norm of the surplus during such an iteration. Let $r_{\min}$ denote the lowest surplus of an agent in S and $r_{\max}$ denote the highest surplus of a type-3 or type-4b agent. Notice that the highest surplus of any agent and hence any agent in S is at most $e \cdot r_{min}$ and that $r_{\max} \le r_{\min}/(1 + 1/n)$.

Each type-1 agent's surplus increases at most by a multiplicative factor of $1 + 1/Rkn^2$. Every type-1 agent partially owns at least one good in $\Gamma(S)$ and therefore, k is at least the number of type-1 agents in $B(S)$. Thus the total increase in the sum of squares of the surpluses as a result of increase in the surpluses of the type-1 agents is $\sum_{b_i \in T_1} (1/Rkn^2) \cdot r_f(b_i)^2$ which is at most $e^2 r_{min}^2 / Rn^2$.

The surplus of the type-2 and the type-3 agents move closer to each other and the decrease in the former is at least as large as the increase in the latter. Therefore, the sum of their surpluses does not increase. We now quantify the decrease in the sum of squares of their surpluses. Let δ_i denote the decrease in the surplus of a type-2 agent b_i and μ_j the increase in surplus of a type-3 agent b_j. Let $r_{f'}$ be the new surplus vector (w.r.t flow f'). Then the change in the sum of squares of type-2, type-3 and type-4 agents is

$$\sum_{b_i \in T_2} ((r_f(b_i) - \delta_i)^2 - r_f(b_i)^2)) + \sum_{b_i \in T_3} ((r_f(b_i) + \mu_i)^2 - r_f(b_i)^2))$$
$$= \sum_{b_i \in T_2} (-2r_f(b_i)\delta_i + \delta_i^2) + \sum_{b_i \in T_3} (2r_f(b_i)\mu_i + \mu_i^2))$$
$$= \sum_{b_i \in T_2} -r_f(b_i)\delta_i + \sum_{b_i \in T_3} r_f(b_i)\mu_i - \sum_{b_i \in T_2} \delta_i(r_f(b_i) - \delta_i) + \sum_{b_i \in T_3} \mu_i(r_f(b_i) + \mu_i)$$

For any balancing iteration we have that $\min_{b_i \in T_2}(r_f(b_i) - \delta_i) \geq \max_{b_i \in T_3}(r_f(b_i) + \mu_i)$ and $\sum_{b_i \in T_2} \delta_i \geq \sum_{b_i \in T_3} \mu_i$. This implies that $\sum_{b_i \in T_2} \delta_i(r_f(b_i) - \delta_i) \geq \sum_{b_i \in T_3} \mu_i(r_f(b_i) + \mu_i)$. Notice that $r_{\min}$ is $\min_{b_i \in T_1 \cup T_2} r_f(b_i)$ and $r_{\max}$ is $\max_{b_i \in T_3 \cup T_{4b}} r_f(b_i)$. Therefore, we may continue

$$\leq -r_{\min} \sum_{b_i \in T_2} \delta_i + r_{\max} \sum_{b_i \in T_3} \mu_i$$
$$\leq -(r_{\min} - r_{\max}) \cdot \sum_{b_i \in T_2} \delta_i \leq -(r_{\min} - r_{\max}) \cdot \frac{(\sum_{b_i \in T_2} \delta_i + \sum_{b_i \in T_3} \mu_i)}{2}.$$

Now, whenever $x = min(x_{23}, x_{24}, x_{13}, x_2)$, $\sum_{b_i \in T_2} \delta_i + \sum_{b_i \in T_3} \mu_i \geq r_{\min} - r_{\max}$. Thus

$$\leq -\frac{(r_{\min} - r_{\max})^2}{2} \leq -\frac{r_{\min}^2}{2(n+1)^2} \leq -\frac{r_{\min}^2}{4n^2}.$$

Therefore $\|r_{f'}\|_2^2 - \|r_f\|_2^2 \leq \frac{e^2 r_{\min}^2}{Rn^2} - \frac{r_{\min}^2}{4n^2} = -\frac{r_{\min}^2}{4n^2}$ (Recall that $R = 8e^2$). Since $\|r_f\|_2^2 \leq ne^2 r_{\min}^2$, we have

$$\|r_{f'}\|_2^2 \leq (1 - \Omega(\frac{1}{n^3}))\|r_f\|_2^2.$$

Now we look into the case when $x = x_{\text{eq}}$. We update the flow exactly the same way as in [10]. The new equality edge will involve a type-1 agent or a type-2 agent. But after we update the flow, we are in a similar situation as above, with a possibility that the surplus of a few type-1 agents may even decrease and be equal to that of a few type-3 agents. Like earlier the sum of surpluses of the agents is non-increasing here too and all arguments for the decrease in the sum of squares of the agents remain the same. The flow adjustment is exactly identical to the one in [10].

▶ **Lemma 10.** *The total number of balancing iterations is* $\mathcal{O}(n^3 \max(n, m) \log(nmUW))$

Proof. Every balancing iteration results in a multiplicative decrease of $1 - \Omega(\frac{1}{n^3})$ in the $L2$ norm of the surplus. The total multiplicative increase as a result of $x_{\max}$-iterations is $(nmUW)^{\mathcal{O}(\max(n.m))}$. Initially $\|r_f\|_2$ is at most $\sqrt{n(mW)^2}$ and the algorithm terminates with $\|r_f\|_2$ being ε. Therefore the total number of balancing iterations is at most

$$\log_{1-\Omega(\frac{1}{n^3})}(\frac{1}{\varepsilon} \cdot \sqrt{n(mW)^2} \cdot (nmUW)^{\mathcal{O}(m)}) = \mathcal{O}(n^3 \max(n, m) \log(nmUW)).$$ ◀

So now we have bounded all the iterations of our algorithm.

▶ **Theorem 11.** *The total number of iterations is* $\mathcal{O}(n^3 \max(n, m) \cdot \log(nWU))$.

2.2.5 Extraction of Equilibrium Prices and Perturbation of Utilities

In [11] is was shown for the special case ($n = m$ and w the identity matrix) that once the total surplus is sufficiently small, the equality network for the current price vector p is the equality network for the equilibrium price vector $\hat{p}$. The equilibrium price vector can then be extracted by solving a linear system. Darwish [4] showed that the same approach also works for the general case.

▶ **Theorem 12.** *Consider any instance of the general linear Arrow-Debreu market with n agents, m goods, utility matrix u and weight matrix w. If p is a price vector such that $\|r_f\|$ at most $1/(8 \cdot (n+m)^{4(n+m)}(UW)^{3(n+m)})$ for any balanced flow f in N_p, then the equilibrium price vector p^* can be determined in $\mathcal{O}((n+m)^4 \cdot \log(UW))$.*

[10] achieves $\mathcal{O}(n^2)$ time for determining the balanced flow in every iteration by keeping the Equality Network acyclic at every point in time in the algorithm. This improvement, after minor adaptions, also applies to the general case. More details are given in the full version of the paper in [2].

2.3 Summary

▶ **Theorem 13.** *The market clearing price vector for the general Arrow-Debreu market can be determined with $\mathcal{O}((n+m)^6 \log(nmWU))$ arithmetic operations.*

Proof. We perturb the utility matrix following the same perturbation as in [10]. This perturbation ensures that the equality network is acyclic at any point in time in the algorithm. Thereafter we run Algorithm 1 until $\|r_f\| < 1/(8 \cdot (n+m)^{4(n+m)}(UW)^{3(n+m)})$. This involves $\mathcal{O}((n)^3 m \log(nWU))$ iterations. In each iteration we can determine x_{23}, x_{24}, x_{13}, x_2 and x_{eq} in $\mathcal{O}(n^2)$ comparisons. The balanced flow can also be determined by $n+m$ max flow calls in N_p. Since N_p is acyclic (due to the perturbation), we can compute each max flow in $\mathcal{O}(n+m)$ arithmetic operations as in [10]. Thus every iteration involves $\mathcal{O}((n+m)^2)$ arithmetic operations and comparisons. Therefore Algorithm 1 terminates performing $\mathcal{O}((n+m)^6 \log(nmWU))$ arithmetic operations and comparisons. Thereafter we perform extraction as in Theorem 12 in time $\mathcal{O}((n+m)^4 \log(nmUW))$ and determine the equilibrium prices for the perturbed utilities. We then determine equilirbium prices corresponding to the original utilities from the equilibrium prices of the perturbed utilities performing $\mathcal{O}(m^3)$ arithmetic operations in $\mathcal{O}(m^4 \log(UW))$ time (details in the full version of the paper in [2]) . Overall we perform $\mathcal{O}((n+m)^6 \log(nmWU))$ arithmetic operations and comparisons. ◀

To achieve the polynomial running time we can follow the same strategy used in [11]. where we restrict the prices and the update factor to powers of $1+1/L$ where L has polynomial bit length (linear in $n+m$). This guarantees that all arithmetic is done on rationals of polynomial bitlength. This can be adapted to the perturbation as well [10].

3 Lower Bounds for the Algorithms in [11, 10]

In this section, we construct non-trivial instances that make the equilibrium prices exponential in U and forces the algorithms in [11, 10] to execute large number of iterations. We construct an I_n comprises of a set of agents $B = \{b_1, b_2, ..., b_n\}$ and a set of goods $G = \{g_1, g_2, ..., g_n\}$, and n is *even.* There is exactly one unit of each good ($W = 1$) and agent b_i only owns one unit of good g_i. We now define the utility matrix as follows: $u_{i,i-1} = U \quad \text{for } 2 \le i \le n$,

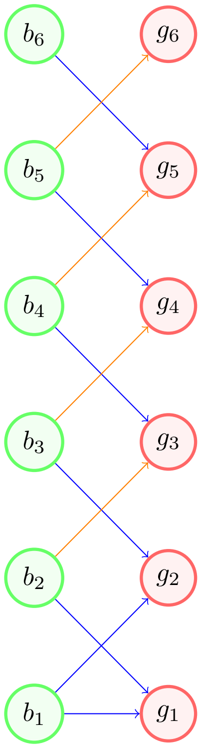

Figure 2 Utility matrix for $n = 6$. The agents are the green nodes on the left and the goods are the red nodes on the right. Agent b_i owns good g_i. The blue edges represent utility U and the orange ones represent utility one.

$u_{i,i+1} = 1 \quad$ for $2 \leq i \leq n-1$, and $u_{1,1} = u_{1,2} = U$. $u_{i,j} = 0$ for every other pair of (i, j). See Figure 2.

▶ **Theorem 14.** *For the instance I_n we have,*

1. *The ratio of the maximum to the minimum price of a good at equilirbium is $\Omega(U^{\Omega(n)})$.*
2. *The algorithms in [11, 10] execute $\Omega(n^4 \log(U))$ iterations.*

Proof. We first show (1). Let p be the market clearing price vector with p_i denoting the price of good g_i, and f be the money flow at equilibrium. Since the only agent interested in g_n is b_{n-1} , $p_{n-1} \geq p_n$. We now discuss two disjoint scenarios,

- $p_n = p_{n-1}$. In this case we claim that for every even i, $p_{i-1} = p_i \geq U \cdot p_{i+1} = U \cdot p_{i+2}$ and $f_{i,i-1} = f_{i-1,i} = p_i = p_{i-1}$. For the base case, $i = n$ we have $p_n = p_{n-1}$ and $f_{n,n-1} = f_{n-1,n} = p_n = p_{n-1}$. For the inductive step we assume that the claim holds for $i+2$. Since g_{i+2} is a bang per buck good for b_{i+1}, $p_i \geq U \cdot p_{i+2} = U \cdot p_{i+1}$. Since the only other agent interested in g_i is b_{i-1} we may conclude that $p_{i-1} \geq p_i$ (b_{i-1} is the only agent investing in g_i). But then again, since the only good b_i invests in is g_{i-1}, $p_{i-1} \leq p_i$. This implies that $p_{i-1} = p_i \geq U \cdot p_{i+1} = U \cdot p_{i+2}$ and $f_{i,i-1} = f_{i-1,i} = p_i = p_{i-1}$.
- $p_{n-1} > p_n$. In this case we claim that for every even i, $p_i < p_{i-1}$ and $p_{i-1} \geq U \cdot p_{i+1}$. For the base case $i = n$, this trivially holds. For the inductive step we assume that our claim holds for $i+2$. Since $p_{i+1} > p_{i+2}$, the agent b_{i+1} must invest in the good g_i. This implies that $p_i \leq U \cdot p_{i+2}$. Since $p_{i+2} < p_{i+1}$, agent b_i must invest in good g_{i+1}. Therefore g_{i+1} is a bang per buck good for agent b_i, implying that $p_{i-1} \geq U \cdot p_{i+1} > U \cdot p_{i+2} \geq p_i$.

In either case, we have $p_{i-1} \geq U p_{i+1}$ for even i. Thus there are goods with price ratio equal to $U^{\frac{n}{2}-1}$. For (2), note that the algorithm in [11] never increases the price of a good more than a multiplicative factor of $(1 + \mathcal{O}(1/n^3))$. Thus the number of iterations is $\log_{1+\mathcal{O}(1/n^3)} U^{\Omega(n)} \in \Omega(n^4 \log(U))$. To prove that the algorithm in [11, 10] takes $\Omega(n^4 \log(U))$ iterations, we first need to understand the details of how the sets S (high surplus agents) and $\Gamma(S)$ (high demand goods) evolve throughout the iterations of the algorithm in [11, 10]. In particular we show that there exists a good g_i and its price is increased by a multiplicative factor of $U^{\Omega(n)}$ in $x_{\max}$ iterations with $\Omega(n)$ type-1 agents. Note that the price of any good is increased at most by a multiplicative factor of $1 + \mathcal{O}(1/n^3)$ in any $x_{\max}$ iteration with $\Omega(n)$ type-1 agents. Since the total price increase in such iterations is $U^{\Omega(n)}$, the number of such iterations is $\Omega(\log_{1+\mathcal{O}(1/n^3)} U^{\Omega(n)}) \in \Omega(n^4 \log(U))$. For the detailed proof we refer to the full version of the paper in [2]. ◀

We next claim that there is an instance I'_n that separates the two algorithms in [11, 10].

▶ **Theorem 15.** *The number of iterations executed by the algorithm in [11] on the instance* I'_n *is* $\Omega((n^{4+\frac{1}{3}} \log(U))/\log(n)^4)$.

We refer to the full version of the paper [2] for the exact construction and the detailed proof of Theorem 15.

References

1. Kenneth J. Arrow and Gérard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22:265–290, 1954. URL: `http://www.jstor.org/stable/1907353`.
2. B. R. Chaudhury and K. Mehlhorn. Combinatorial Algorithms for General Linear Arrow-Debreu Markets. *ArXiv e-prints*, October 2018. `arXiv:1810.01237`.
3. Bruno Codenotti, Benton McCune, and Kasturi R. Varadarajan. Market equilibrium via the excess demand function. In *STOC*, pages 74–83. ACM, 2005.
4. Omar Darwish. Market Equilibrium Computation for the Linear Arrow-Debreu Model. Master's thesis, Fachbereich Informatik, Saarland University, 2016.
5. Omar Darwish and Kurt Mehlhorn. Improved Balanced Flow Computation Using Parametric Flow. *Information Processing Letters*, pages 560–563, 2016.
6. Omar Darwish and Kurt Mehlhorn. Linear Arrow-Debreu Markets: The General Case. Unpublished, 2016.
7. Nikhil R. Devanur, Jugal Garg, and László A. Végh. A Rational Convex Program for Linear Arrow-Debreu Markets. arXiv:1307.803, 2013.
8. Nikhil R. Devanur, Christos H. Papadimitriou, Amin Saberi, and Vijay V. Vazirani. Market equilibrium via a primal–dual algorithm for a convex program. *J. ACM*, 55(5):22:1–22:18, November 2008. `doi:10.1145/1411509.1411512`.
9. Nikhil R. Devanur and Vijay V. Vazirani. An improved approximation scheme for computing Arrow-Debreu prices for the linear case. In *FSTTCS*, pages 149–155, 2003.
10. Ran Duan, Jugal Garg, and Kurt Mehlhorn. An improved combinatorial algorithm for linear Arrow-Debreu markets. In *SODA*, pages 90–106, 2016.
11. Ran Duan and Kurt Mehlhorn. A Combinatorial Polynomial Algorithm for the Linear Arrow-Debreu Market. *Information and Computation*, 243:112–132, 2015. a preliminary version appeared in ICALP 2013, LNCS 7965, pages 425-436.
12. B. Curtis Eaves. A Finite Algorithm for the Linear Exchange Model. *Journal of Mathematical Economics*, 3:197–203, 1976.

13 Jugal Garg, Ruta Mehta, Milind Sohoni, and Vijay V. Vazirani. A Complementary Pivot Algorithm for Markets Under Separable, Piecewise-linear Concave Utilities. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1003–1016, New York, NY, USA, 2012. ACM. doi:10.1145/2213977.2214068.

14 Rahul Garg and Sanjiv Kapoor. Auction Algorithms for Market Equilibrium. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 511–518, New York, NY, USA, 2004. ACM. doi:10.1145/1007352.1007430.

15 Kamal Jain. A Polynomial Time Algorithm for Computing an Arrow-Debreu Market Equilibrium for Linear Utilities. *SIAM J. Comput.*, 37(1):303–318, April 2007. doi:10.1137/S0097539705447384.

16 Kamal Jain, Mohammad Mahdian, and Amin Saberi. Approximating Market Equilibria, 2003.

17 E. I. Nenakov and M. E. Primak. One algorithm for finding solutions of the Arrow-Debreu model. *Kibernetica*, 3:127–128, 1983.

18 A. Wald. Über einige Gleichungssysteme der mathematischen Ökonomie. *Zeitschrift für Nationalökonomie*, 7:637–670, 1936. Translated: Econometrica, Vol. 19(4), p.368–403, 1951.

19 Léon Walras. *Elements of Pure Economics, or the theory of social wealth.* Porcupine Press, 1874.

20 Yinyu Ye. A path to the Arrow-Debreu competitive market equilibrium. *Math. Program.*, 111(1):315–348, June 2007. doi:10.1007/s10107-006-0065-5.