An Auction Algorithm for Market Equilibrium with Weak Gross Substitute Demands

JUGAL GARG, University of Illinois at Urbana-Champaign, USA
EDIN HUSIĆ, IDSIA, USI-SUPSI, Switzerland
LÁSZLÓ A. VÉGH, London School of Economics and Political Science, UK

We consider the Arrow–Debreu exchange market model under the assumption that the agents' demands satisfy the weak gross substitutes (WGS) property. We present a simple auction algorithm that obtains an approximate market equilibrium for WGS demands assuming the availability of a price update oracle. We exhibit specific implementations of such an oracle for WGS demands with bounded price elasticities and for Gale demand systems.

CCS Concepts: • **Theory of computation** → **Market equilibria**.

Additional Key Words and Phrases: Auction algorithm, Weak gross substitutes, Fisher equilibrium, Gale equilibrium

ACM Reference Format:
Jugal Garg, Edin Husić, and László A. Végh. 2025. An Auction Algorithm for Market Equilibrium with Weak Gross Substitute Demands. *ACM Trans. Econ. Comput.* 0, 0, Article 0 (May 2025), 24 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Market equilibrium is a fundamental model in mathematical economics to describe the balance between supply and demand. The study of market equilibria was pioneered by Walras [63] in 1874, and was developed in the 1950s by Arrow and Debreu [4] and McKenzie [55]. In this paper, we focus on the classical exchange market setting, where a set of agents A arrives at the market with initial endowments of infinitely divisible goods G. A market equilibrium comprises prices for the goods and a fractional assignment between the goods and the agents. Prices and assignments form a *market equilibrium* if (a) each agent receives a bundle of goods they prefer the most at the given prices by spending their revenue from selling their initial endowment, and (b) the market clears: the demand of each good meets its supply. A typical way to represent the preferences is by utility functions $u_i : \mathbb{R}_+^G \to \mathbb{R}_+$ for each agent $i \in A$; the demand of agent i at prices p and revenue b_i is a bundle x_i maximizing $u_i(x_i)$ subject to $\langle p, x_i \rangle \leq b_i$. Classical works by Arrow and Debreu [4]

A preliminary version of this paper appeared in the Proceedings of the 38th Symposium on Theoretical Aspects of Computer Science (STACS 2021) [37].
JG was supported by an NSF Grant CCF-1942321. EH was supported by SNSF Grant 200021 200731/1. LV received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. ScaleOpt–757481).
Authors' addresses: Jugal Garg, University of Illinois at Urbana-Champaign, Urbana, USA, jugal@illinois.edu; Edin Husić, IDSIA, USI-SUPSI, Switzerland, edin.husic@idsia.ch; László A. Végh, London School of Economics and Political Science, London, UK, l.vegh@lse.ac.uk.

2167-8375/2025/5-ART0 $15.00
https://doi.org/0000001.0000001

and McKenzie [55, 56] showed the existence of market equilibrium under mild assumptions, using Kakutani's fixed point theorem.

Equilibrium constitutes the ideal limit behavior of markets; existence proofs based on fixed point theorems do not explain how such a limit can be attained. Investigating market dynamics has been an important topic since the early days: Walras [63] introduced the *tâtonnement* process, a natural dynamics of supply and demand. This can be seen as a multi-round auction process, where an auctioneer announces the current prices in each round. At these prices, each agent submits their most preferred bundle of goods. Prices are adjusted in light of these bids: prices of overdemanded goods are increased, and prices of underdemanded goods are decreased.

Samuelson [60] formulated a continuous version of tâtonnement as a dynamical system. Works by Arrow and Hurwitz [6], Arrow, Block, and Hurwitz [3] introduced the *weak gross substitutability (WGS)* property as a sufficient condition for convergence to an equilibrium. The agent's demands are said to be WGS if the demand for any good does not increase when its price increases while the rest of the prices remain unchanged. Such a property gives a sound justification of the tâtonnement price changes. However, it is a nontrivial requirement, and there are important examples of demands that are not WGS. Scarf [61] showed that tâtonnement may not converge for non-WGS demands.

Classical market equilibrium models became the subject of renewed interest in the optimization and theoretical computer science communities, starting from the 1991 paper by Megiddo and Papadimitriou [57]. Formulating the existence of market equilibrium as a computational search problem raises intriguing questions. Many of these works treat market equilibrium as a centrally coordinated computational problem: can a central authority compute a market equilibrium given perfect information on all agents' utilities? Surprisingly, even in such a centrally coordinated setting, and even for some of the simplest non-WGS demands, computing approximate equilibria turn out to be complete problems for certain complexity classes. On the positive side, this line of investigation led to remarkable algorithmic developments for various market models. See e.g., [12, 16, 22, 24, 29, 32, 39, 40, 48, 62, 65]. For WGS utilities, the first polynomial-time computability of market equilibria was established by Codenotti, Pemmaraju, and Varadarajan [23]. A discrete variant of the tâtonnement algorithm that converges to an approximate equilibrium (see also [59, Section 6.3]) was given by Codenotti, McCune, and Varadarajan [21]. More recently, Bei, Garg, and Hoefer [8] gave a simple ascending-price algorithm. We note that both the latter algorithms need central coordination.

In most market settings, we cannot assume the level of central coordination needed for many of the above algorithms. In such markets, one has to investigate distributed mechanisms in a decentralized environment with limited coordination. This issue was addressed by a number of papers providing tâtonnement algorithms for various classes of utility functions and restricted models, some of them substantially weakening the need for central coordination among agents, see e.g., [7, 17, 18, 26, 35].

Auction algorithms form an even simpler subclass of tâtonnement-type algorithms. These algorithms are decentralized and require only local coordination between agents. Each agent may take goods from others by outbidding them, i.e., offering slightly higher prices. While prices in tâtonnement may increase as well as decrease, prices in auction algorithms may only go up. For exchange market models, the first such algorithm was established for linear utilities—of the form $u(x) = \sum_{j\in G} v_j x_j$—by Garg and Kapoor [41] (see also [59, Section 5.12]). The algorithm was later improved [42] and generalized to separable concave gross substitute utility functions [44], to a subclass of non-separable gross-substitutes called *uniformly separable* [43], and to a production model with linear production constraints and linear utilities [49].

Auction algorithms have been widely used beyond exchange markets and studied in different contexts in optimization and economics. Bertsekas [9, 10] introduced auction algorithms for assignment and transportation problems. Closely related algorithms were introduced for markets with indivisible goods—further discussed in Section 1.2—by Kelso and Crawford [51], and Demange, Gale, and Sotomayor [28].

1.1 Our contributions

We present a new auction algorithm that computes an approximate market equilibrium in exchange markets for arbitrary WGS utilities, assuming a suitable oracle representation. This settles an open question raised in [43]. The result affirms the natural intuition that the WGS property should suffice for auction algorithms: A main invariant in auction algorithms is that at every price increase, the agents will still hold on to the goods whose prices have not increased. This property is almost identical to the definition of the WGS property; nevertheless, making an auction algorithm work for general WGS utilities requires new technical ideas.

The previously mentioned auction algorithms operate with two prices for each good j, a lower price p_j and a higher price $(1+\varepsilon)p_j$. This technique was used for linear [41], for separable [42], and for uniformly separable utilities [44]. However, this simple approach does not seem to apply to the general WGS case, and we need to use a more fine-grained pricing approach. In our algorithm, each agent maintains individual prices for each good j in the range $[p_j, (1+\varepsilon)p_j]$ for the 'market price' p_j. The main invariant in our algorithm is that each agent maintains a subset of an optimal bundle with respect to these individual prices.

Each agent updates their individual prices using a subroutine called `FindNewPrices`. The general algorithm in Section 3 relies on this subroutine, and its running time for finding an ε-approximate equilibrium is $O\left(\frac{nmT_F}{\varepsilon^2} \cdot \log\left(\frac{p_{\max}}{p_{\min}}\right)\right)$, where T_F denotes a running time bound on `FindNewPrices` and $p_{\max}$ and $p_{\min}$ are lower and upper bounds on the prices in an approximate equilibrium (Theorem 1).

Demand systems. Our algorithm uses only local coordination between agents. However, agents should update their individual prices according to certain requirements. These are captured by `FindNewPrices`; implementing this subroutine depends on the particular demand system.

First, one needs to clarify how the preferences are represented in the model. WGS utilities in the literature are usually given in an explicit form such as CES (constant elasticity of substitution) or Cobb-Douglas utilities, see Section 2.2. This is in contrast with the setting of markets with indivisible goods, where the common model is via a value or demand oracle [52], since direct preference elicitation, that is, the explicit description of the valuation function would be exponential. The class of continuous WGS functions is also rich, and hence an oracle approach seems more appropriate to devise algorithms for this class.

We model the agent preferences by *demand oracles* (Definition 5). A demand oracle may be implemented by solving a utility-maximizing convex problem but may be of a different form. We discuss this in Section 2.2, also exhibiting a class of WGS demand systems where our model is applicable but do not appear to have a simple closed-form representation.

A natural parametrization of WGS demand systems is by *price elasticity* (Definition 6) that bounds the change in the demands as a function of the price changes. In Section 4.1, we implement `FindNewPrices` by a simple iterative application of the demand oracle for the case of bounded price elasticities.

We present additional implementations for the case when the price elasticity can be unbounded. Linear utilities constitute an important such class. Lemma 10 gives a direct, linear time implementation of `FindNewPrices` for linear utilities.

In Section 4.3, we consider Gale demand systems introduced by Nesterov and Shikhman [58]. Such demand systems are given by a convex program. Accordingly, we use a convex programming approach to implement `FindNewPrices`.

Spending restricted equilibrium and Nash social welfare. Our motivation for considering Gale demand systems comes from an application to the *Nash social welfare (NSW)* problem. In this problem, we need to allocate a set of indivisible goods to agents in order to maximize the geometric mean of their valuations.

A useful relaxation of the NSW problem turns out to be a so-called *spending-restricted (SR) equilibrium* under Gale demand systems. Spending-restricted equilibria were introduced by Cole and Gkatzelis [27] as a key tool in finding the first constant-factor approximation algorithm for this problem with additive valuations. The same equilibrium concept was used in several other approximation algorithms for the NSW problem, see e.g., [1, 15, 25, 36].

The SR-equilibrium is a variant of the *Fisher market model*, a special case of the exchange market model. In the Fisher model, the agents do not arrive with an initial endowment of goods but with a fixed budget to spend on the available set of goods. The SR-equilibrium differs in that the available amount of each good j is influenced by the price p_j, namely, it is set as $\min\{1, 1/p_j\}$. In other words, once the price of the good reaches 1, the seller will only sell an amount of total value 1. Auction algorithms are well-suited for SR-equilibrium computation: once the price of a good goes above one, we can naturally decrease the total available amount of these goods within the auction framework.

In an extended version of this paper [38], we design a polynomial time constant-factor approximation algorithm for the NSW problem under capped separable piecewise linear concave valuations by rounding an SR equilibrium under Gale demand systems to an approximately optimal solution. The previous algorithm for this problem takes pseudopolynomial time [15]. A key for this result is finding an approximate SR equilibrium in polynomial-time for which we use a modification of our auction algorithm. Interestingly, the capped separable piecewise linear concave valuations satisfy the WGS property under Gale demand systems but not in the "standard" demand system setting (1). For details, we refer to [37, 38].

1.2 Further related work

Proportional response dynamics. Proportional response is a distributed market mechanism introduced by Zhang [66] in the context of Fisher markets. In contrast with tâtonnement and auctions, there is no direct price mechanism. In each round, agents bid on goods in proportion to the utility they receive from them in the previous round; the goods are then allocated in proportion to the agents' bids. Proportional response is known to converge to a market equilibrium in a variety of Fisher markets [11, 19, 20], and some special cases of exchange markets [13, 14, 64].

Markets with indivisible goods. Auction algorithms have been widely studied in the context of markets with *indivisible goods*. There are significant differences between the settings with divisible and indivisible goods. In the indivisible setting, equilibria are known to exist only in restricted settings. Kelso and Crawford [51] introduced (discrete) gross substitute utilities as a class where an equilibrium is guaranteed to exist, and a simple auction algorithm can be used to find an approximate equilibrium. As shown by Gul and Stacchetti [46, 47], the discrete gross substitutes property is, in essence, a necessary and sufficient condition for the existence of an equilibrium and for an auction algorithm to work. We refer the reader to the survey by Paes Leme [52] on the role of gross substitute utilities in markets with indivisible goods and their connections to discrete convex analysis.

Whereas the definitions of discrete gross substitutes and continuous WGS utilities are very similar, there does not appear to be a direct connection between these notions. The main difference is in the utility concepts: for indivisible markets, the standard model is to maximize the valuation minus the price of the set at given prices, whereas standard divisible market models operate with *fiat money*: the prices appear via the budget constraints but not in the utility value. Still, our result can be interpreted as the continuous analogue of the strong link between auction algorithms and the gross substitutes property for markets with indivisible goods: we show that auction algorithms are applicable for the entire class of WGS utilities for markets with divisible goods. We suspect that the converse may also be true, namely, that the applicability of auction algorithms should be limited to WGS utilities. In contrast, tâtonnement algorithms have been successfully applied beyond the WGS class, see e.g., [17, 18, 35].

Graphical exchange economies. Subsequently to the preliminary version of this work [37], Andrade, Frongillo, Gorokhovsky, and Srinivasan [2] studied graphical exchange markets with resale. Here, agents may only trade with their neighbors in a graph. They show the existence of such an equilibrium and give an auction algorithm for finding an approximate market equilibrium in such markets, assuming that agents have WGS demands.

The rest of the paper is structured as follows. Section 2 defines the exchange market model and provides examples of WGS demand systems. Section 3 presents the auction algorithm for exchange markets. Section 4 present different ways of implementing `FindNewPrices`—the key subroutine of the algorithm. Section A compares the running time of our algorithm to previous work.

A preliminary version of this paper appeared in [37], and an extended version, also including results on SR-equilibrium can be found in [38].

2 THE EXCHANGE MARKET AND DEMAND SYSTEMS

We use $\mathbb{R}_+$ for the nonnegative reals, and for a positive integer k, let $[k] = \{1, 2, \dots, k\}$. We consider a market with a set of agents $A = [n]$ and a set of divisible goods $G = [m]$. Each agent $i \in [n]$ arrives at the market with an initial endowment of goods $e^{(i)} \in \mathbb{R}_+^m$. We let $e = \sum_{i=1}^n e^{(i)}$ denote the total amount of the goods. We assume $e_j > 0$ for each $j \in [m]$. A *bundle* x is a non-negative vector $x \in \mathbb{R}_+^m$. We say that a bundle of goods $y \in \mathbb{R}_+^m$ *dominates* the bundle $x \in \mathbb{R}_+^m$ if $x \leq y$.

Given a non-negative price vector $p \in \mathbb{R}_+^m$, the budget of agent i at prices p is defined as $b_i(p) = \left\langle p, e^{(i)} \right\rangle$; we simply write b_i if the prices are clear from the context. It follows that $\langle p, e \rangle = \sum_{i=1}^n \left\langle p, e^{(i)} \right\rangle = \sum_{i=1}^n b_i$.

We specify the markets via *demand systems.* A *demand system* is a function $D : \mathbb{R}_+^{m+1} \to 2^{\mathbb{R}_+^m}$; $D(p, b)$ denotes the set of preferred bundles of an agent at prices p that are affordable within budget b. Here $2^{\mathbb{R}_+^m}$ denotes the family of all subsets of $\mathbb{R}_+^m$. Bundles in $D(p, b)$ are called *optimal bundles* or *demand bundles* at prices p and budget b. The demand system is *simple* if $|D(p, b)| = 1$ for all $(p, b) \in \mathbb{R}^{m+1}$; for such demand systems, we will also use $D(p, b)$ to denote this single bundle. We make two assumptions on the demand systems.

Assumption 1 (Scale invariance)**.** *For every agent $i \in [n]$, $(p, b) \in \mathbb{R}_+^{m+1}$, and $\alpha > 0$, $D_i(p, b) = D_i(\alpha p, \alpha b)$.*

That is, we require that the demand is homogeneous of degree 0; informally, the demand does not depend on the currency. This is a standard assumption in microeconomics, see e.g., [5, 31, 33, 54].

Assumption 2 (Non-satiation)**.** *For every agent $i \in [n]$, $(p, b) \in \mathbb{R}_+^{m+1}$, and every $x \in D_i(p, b)$, $\langle p, x \rangle = b$.*

That is, in every optimal bundle the agents must fully spend their budgets. This is a standard assumption for exchange markets as it is necessary for the fundamental theorems of welfare economics (see e.g., [53, Chapter 16]).[1]

A common way to define demand systems is by utility functions. By a *utility function* we mean a function $u : \mathbb{R}^m_+ \to \mathbb{R}_+$ that is concave, continuous, non-decreasing with $u(0) = 0$. The corresponding demand system is

$$D^u(p, b) := \arg\max_{x \in \mathbb{R}^m_+} \{u(x) : \langle p, x \rangle \leq b\} \ . \tag{1}$$

An important example is the *linear demand system* defined by a linear utility function $u(x) = \langle v, x \rangle$ for $v \in \mathbb{R}^m_+$. The corresponding demand system $D^u(p, b)$ is the set of all fractional assignments of goods maximizing v_j/p_j (bang-per-buck) with a total cost of b. Thus, this demand system is not simple.

For demand systems in the form (1), Assumption 1 is immediate, and Assumption 2 holds if $u(x)$ is strictly monotone increasing. The demand system is simple if $u(x)$ is strictly convex.

2.1 Exact and approximate market equilibria

Definition 1 (Market equilibrium)**.** *Consider an exchange market with a set $A = [n]$ of agents, a set $G = [m]$ of goods, initial endowments $e^{(i)} \in \mathbb{R}^m_+$, and $e = \sum_{i=1}^n e^{(i)}$. Let $D_i(p, b)$ denote the demand system of agent $i \in A$. The prices $p \in \mathbb{R}^m_+$ and bundles $x^{(i)} \in \mathbb{R}^m_+$ form a* market equilibrium *if*

(i) $x^{(i)} \in D_i(p, \langle p, e^{(i)} \rangle)$ for all $i \in A$, and

(ii) $\sum_{i=1}^n x_j^{(i)} \leq e_j$, with equality whenever $p_j > 0$, for all $j \in G$.

That is, p and the optimal bundles $x^{(i)}$ form an equilibrium if no good is overdemanded and goods at a positive price are fully sold. Note that this implies that every agent fully spends their budget.

We relax this to the following notion of ε-approximate equilibrium:

Definition 2 (Approximate market equilibrium)**.** *Consider the same setting as in Definition 1. For $\varepsilon \geq 0$, the prices $p \in \mathbb{R}^m_+$ and bundles $x^{(i)} \in \mathbb{R}^m_+$ form an ε-*approximate market equilibrium *if*

(i) $x^{(i)} \leq z^{(i)}$ for some $z^{(i)} \in D_i(p^{(i)}, \langle p, e^{(i)} \rangle)$, where $p \leq p^{(i)} \leq (1 + \varepsilon)p$,

(ii) $\sum_{i=1}^n x_j^{(i)} \leq e_j$, and

(iii) $\langle p, e - \sum_{i=1}^n x^{(i)} \rangle \leq \varepsilon \langle p, e \rangle$.

That is, every agent receives a subset of their optimal bundle at prices that are within a factor $(1+\varepsilon)$ from p, and all goods are nearly sold: the value of the unsold goods is at most an ε fraction of the total value of the goods. The total value of the goods "taken away" from the near-optimal bundles of the agents is $\sum_{i=1}^n \langle p, z^{(i)} - x^{(i)} \rangle$. Parts (i) and (iii), together with the fact that $\langle p^{(i)}, z^{(i)} \rangle \leq \langle p, e^{(i)} \rangle$ for all i, imply that this amount is at most $\varepsilon \langle p, e \rangle$. In particular, $\varepsilon = 0$ corresponds to an exact market equilibrium as in Definition 1.

Condition (i) can be seen as a natural extension of the corresponding approximate optimality conditions in previous auction algorithms [41, 43, 44]. For linear utilities, Garg and Kapoor [41] require the approximate maximum bang-per-buck condition $v_{ij}/p_j \leq (1 + \varepsilon)v_{ik}/p_k$ for any agent i, goods j and k such that $x_{ik} > 0$. In other words, the goods purchased by agent i according to this definition are maximum bang-per-buck with respect to some prices $p^{(i)}$ such that $p \leq p^{(i)} \leq (1+\varepsilon)p$.

Condition (iii) corresponds to the definition of approximate equilibrium in [30] and [45]. This notion is weaker than the ones used in [41, 43, 44]. The most important difference is that the latter

[1]We note that this assumption can be replaced by a weaker one in the case of Fisher markets, see [37, 38].

papers guarantee that each agent recovers approximately their optimal utility. Such a property could be achieved by strengthening the bound in (iii) from $\varepsilon \langle p, e \rangle$ to $\varepsilon p_{\min} e_{\min}$, where $p_{\min}$ is the minimum price and $e_{\min}$ is the smallest total fractional amount in the initial endowment of any agent. However, this would come at the expense of substantially worse running time guarantees in our algorithmic framework.

An important special case of exchange markets are *Fisher markets*, where $e^{(i)} = \frac{b_i}{\sum_{i=1}^n b_i} e$ for each $i \in [n]$, where $b_i > 0$. That is, the initial endowments include every good in the same proportion. By appropriately scaling the prices, we can interpret the b_i's as fixed budgets, and an exchange market equilibrium can be written as follows.

Definition 3 (Fisher market equilibrium). *Consider a Fisher market with a set $A = [n]$ of agents, a set $G = [m]$ of goods, and budgets $b_i > 0$, $i \in [n]$. Let $D_i(p, b)$ denote the demand system of agent $i \in A$. The prices $p \in \mathbb{R}_+^m$ and bundles $x^{(i)} \in \mathbb{R}_+^m$ form a* Fisher market equilibrium *if*

(i) $x^{(i)} \in D_i(p, b_i)$ for all $i \in A$, and
(ii) $\sum_{i=1}^n x_j^{(i)} \le e_j$, with equality whenever $p_j > 0$, for all $j \in G$.

2.2 The weak gross substitutes property

We next introduce the class of demand systems investigated in this paper.

Definition 4. *The demand system $D(p, b)$ is a* weak gross substitutes (WGS) demand system *if for any $(p, b) \in \mathbb{R}_+^{m+1}$, any $x \in D(p, b)$, and any $p' \ge p$ and $b' \ge b$, there exists $y \in D(p', b')$ such that $y_j \ge x_j$ whenever $p'_j = p_j$.*

Further, we say that the utility function $u : \mathbb{R}_+^m \to \mathbb{R}_+$ satisfies the WGS property if the corresponding demand system $D^u(p, b)$ as in (1) *is a WGS demand system.*

We use an oracle model to represent the demand systems. We require access to the allocations guaranteed by Definition 4.

Definition 5 (Demand oracle). *For a WGS demand system $D(p, b)$, a* WGS demand oracle *requires in the input two vectors $(p, b), (p', b') \in \mathbb{R}_+^{m+1}$ such that $(p', b') \ge (p, b)$, and a vector $x \in D(p, b)$. The oracle outputs a vector $y \in D(p', b')$ such that that $y_j \ge x_j$ whenever $p'_j = p_j$.*

The complex form of the definition is due to the possible non-uniqueness of demand bundles. For simple demand systems, it suffices to specify $(p', b') \in \mathbb{R}_+^{m+1}$ in the input; the output is the unique vector $D(p', b')$.

Consider a demand system $D^u(p, b)$ as in (1) for a utility function $u : \mathbb{R}_+^m \to \mathbb{R}_+$. If this is not a simple demand system, we can implement the demand oracle by adding the constraints $y_i \ge x_i$ for every i with $p'_i = p_i$ to the convex optimization problem in (1).

Examples of WGS utilities. Some classical examples in the literature are as follows.

- As previously mentioned, the *linear utility function* is given by $u(x) = \langle v, x \rangle$ for $v \in \mathbb{R}_+^m$.
- The *Cobb-Douglas* utility function is specified by parameters $\alpha \in \mathbb{R}_+^m$, $\sum_{j=1}^m \alpha_j = 1$ as

$$u(x) := \prod_{j=1}^m x_j^{\alpha_j} .$$

This is a simple demand system with $x = D^u(p, b)$ such that $x_j = b\alpha_j / p_j$ for all $j \in [m]$.

- The *constant elasticity of substitution (CES)* utility function is specified by parameters $\beta \in \mathbb{R}^m_+$ such that $\sum_{j=1}^m \beta_j = 1$, and $\sigma \in \mathbb{R}_+$ as
$$u(x) := \left(\sum_{j=1}^m \beta_j^{\frac{1}{\sigma}} x_j^{\frac{\sigma-1}{\sigma}}\right)^{\frac{\sigma}{\sigma-1}},$$
This is also a simple demand system with $x = D^u(p,b)$ such that $x_j = \dfrac{\beta_j p_j^{-\sigma} b}{\sum_{k=1}^m \beta_k p_k^{1-\sigma}}$ for all $j \in [m]$. The CES demand system satisfies the WGS property if and only if $\sigma > 1$.
- The *nested CES* utility function is defined recursively (see [48] for more details). Any CES function is a nested CES function. If $g, h_1, \ldots, h_t$ are nested CES functions, then $f(x) = \max g(h_1(x^1), \ldots, h_t(x^t))$ over all $x^1, \ldots, x^t$ such that $\sum_{k=1}^t x^k = x$, is a nested CES function. In a well-studied special case, each good j can only be used in at most one of the h_i's; see e.g., [50].

Convex combinations of demand systems. Given two WGS utility functions u and u', the demand system corresponding to their sum $u + u'$ may not be WGS. In contrast, taking convex combinations of simple WGS demand systems retains this property; the following proposition is easy to verify.

Proposition 1. *Let $D(p,b)$ and $D'(p,b)$ be two simple WGS demand systems and $0 \le \lambda \le 1$. Let us define the demand system $D'' = \lambda D + (1-\lambda)D'$ by*
$$D''(p,b) := D(p, \lambda b) + D'(p, (1-\lambda)b)\,.$$
Then, D'' is a simple WGS demand system.

This enables the construction of some interesting demand systems. For example, Matsuyama and Ushchev [54] consider hybrids of CES and Cobb-Douglas demands, where the demand system can be given as
$$x_j = \frac{b}{p_j}\left[\lambda\alpha_j + (1-\lambda)\frac{\beta_j p_j^{1-\sigma}}{\sum_k \beta_k p_k^{1-\sigma}}\right],$$
for $\beta \in \mathbb{R}^m_+$, $\sum_{j=1}^m \beta_j = 1$, $\sigma > 1$, $\alpha \in \mathbb{R}^m_+$, $\sum_{j=1}^m \alpha_j = 1$, and $0 \le \lambda \le 1$.[2]

Note that if $D = D^u$ and $D' = D^{u'}$ for some concave utility functions u and u', the demand system $\lambda D + \lambda' D'$ will in general not correspond to the utility function $\lambda u + \lambda' u'$. It is not clear whether one can explicitly write utility functions corresponding to such convex combinations.

Using a demand oracle model, our algorithm is applicable to a convex combination of simple demand oracles.

Separable and uniformly separable WGS utility functions. The auction algorithm for linear utilities [41] was later extended to separable WGS utility functions [44], that is, $u = \sum_{j \in G} u_j$ where each u_j is a WGS utility function depending only on good j. This model was further generalized to *uniformly separable* WGS utility functions [43], that is, $\frac{\partial u(x)}{\partial x_j} = f_j(x_j)g(x)$, where each f_j is a strictly decreasing function. This class already includes CES and Cobb-Douglas utilities; however, it does not appear to extend to demand systems obtained as their convex combinations, where even the explicit form of the utility function is unclear. Further, the running time bound stated in [43] is unbounded for the CES and Cobb-Douglas cases; see Section A for further discussion.

[2]We note that this demand function does not seem to correspond to a nested CES utility function.

2.3 Price elasticity of demands

A commonly studied property of demand systems is *price elasticity*. For simple demand systems that are differentiable, the usual definition of the price elasticity of good j with respect to the price of good k is $e_{j,k} = \partial \log D_j(p,b)/\partial \log p_k$, where $D_j(p,b)$ is the unique demand for good j at prices p and budget b. The WGS property guarantees that $e_{j,k} \geq 0$ if $j \neq k$, and consequently, $e_{k,k} \leq 0$.

The following definition does not assume simplicity or differentiability of the demand system. It corresponds to $e_{k,k} \geq -f$ for all $k \in [m]$, in the above case.

Definition 6. *Consider a WGS demand system $D(p,b)$. For some $f > 0$, we say that the* elasticity of $D(p,b)$ is at least $-f$ *if the following holds. For any $(p,b) \in \mathbb{R}_+^{m+1}$ and $x \in D(p,b)$, $j \in [m]$ and $\mu \geq 1$, let us define*

$$p'_k = \begin{cases} \mu p_k & \text{if } k = j\,, \\ p_k & \text{otherwise.} \end{cases} \tag{2}$$

Then, there exists a bundle $x' \in D(p',b)$ such that $x'_j \geq x_j/\mu^f$ and $x'_k \geq x_k$ for every $k \neq j$.

It is easy to check that the linear demand systems do not satisfy this property for any finite f, as the demand for a good may drop to zero as a result of an arbitrarily small price increase. We include the proof of the following well-known statement to illustrate this concept.

Lemma 1. *The Cobb-Douglas demand system has elasticity at least -1, and the CES demand system with parameter $\sigma > 1$ has elasticity at least $-\sigma$.*

Proof. The optimal bundle for a Cobb-Douglas utility function is $x = D(p,b)$ with $x_\ell = b\alpha_\ell/p_\ell$ for $\ell \in [m]$. Increasing the price of a good by a factor $\mu \geq 1$ corresponds to a decrease in the demand by the same factor. Thus, the elasticity is at least -1.

The optimal bundle for CES utilities is $x = D(b,p)$ with $x_\ell = \dfrac{\beta_\ell p_\ell^{-\sigma} b}{\sum_{k=1}^m \beta_k p_k^{1-\sigma}}$ for $\ell \in [m]$. Select any good $j \in [m]$ and $\mu \geq 1$, and let p' be defined as in (2). Let $x' = D(p',b)$ denote the optimal bundle. Using $\sigma > 1$, we get

$$x'_j = \frac{\beta_j \mu^{-\sigma} p_j^{-\sigma} b}{\sum_{k\neq j} \beta_k p_k^{1-\sigma} + \beta_j \mu^{1-\sigma} p_k^{1-\sigma}} = \frac{\beta_j p_j^{-\sigma} b}{\sum_{k\neq j} \beta_k \mu^{\sigma} p_k^{1-\sigma} + \beta_j \mu p_k^{1-\sigma}} > \frac{\beta_j p_j^{-\sigma} b}{\mu^\sigma \sum_k \beta_k p_k^{1-\sigma}} = \frac{x_j}{\mu^\sigma}\,,$$

verifying that the CES demand system has elasticity at least $-\sigma$. □

Our next lemma allows us to derive price elasticity bounds for convex combinations of simple demand systems.

Lemma 2. *Let D and D' be simple demand systems with elasticity at least $-f$ and $-f'$, respectively. Let $0 \leq \lambda \leq 1$. Then the demand system $\lambda D + (1-\lambda)D'$ has elasticity at least* $\min\{-f,-f'\}$.

Proof. Let $D'' = \lambda D + (1-\lambda)D'$ and $f'' = \max\{f,f'\}$. Let $(p,b) \in \mathbb{R}_+^{m+1}$, $x = D(p,\lambda b)$ and $x' = D''(p,(1-\lambda)b)$. Then, $x'' = x + x' = D'(p,b)$.

Let $j \in [m]$, $\mu \geq 1$ and define p' as in (2). As the elasticity of D is at least $-f$, for $y = D(p',\lambda b)$ we have $y_j \geq x_j/\mu^f \geq x_j/\mu^{f''}$. Analogously, for $y' = D'(p',(1-\lambda)b)$ we have $y'_j \geq x'_j/\mu^{f''}$. Thus, $y_j + y'_j \geq (x_j + x'_j)/\mu^{f''}$. Since $y + y' = D''(p,b)$, we conclude that the elasticity of D'' is at least $-f''$. □

2.4 Gale demand systems

Recall that for a utility function $u : \mathbb{R}_+^m \to \mathbb{R}_+$, we can obtain demand systems from utilities using the convex program (1) that maximizes the utility subject to the budget constraint.

Fisher market equilibria can be formulated by the well-known Eisenberg–Gale convex program [34] for many important cases:

$$\max \sum_{i=1}^{n} b_i \log u_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{n} x_i \le e\,. \tag{3}$$

Eisenberg [33] showed that the optimal solutions to this program, together with the prices corresponding to the optimal Lagrangian multipliers, form a Fisher market equilibrium if the utilities are homogeneous of degree one—that is, $u_i(\alpha x) = \alpha u_i(x)$ for every $x \in \mathbb{R}^m$ and $\alpha > 0$, $i \in [n]$. This class includes many important examples such as linear, Cobb–Douglas, and CES utilities.

Nevertheless, solutions to (3) may not correspond to a Fisher market equilibrium in general. Nesterov and Shikhman [58] showed, using Lagrangian duality, that the optimal solutions to (3) always form a Fisher market equilibrium for *Gale demand systems* defined as:

$$G^u(p,b) := \operatorname*{argmax}_{x\in\mathbb{R}_+^m}\; b\log u(x) - \langle p, x\rangle\;. \tag{4}$$

The following connection explains why the Eisenberg–Gale program can be used for demand systems of the form (1) for homogeneous degree one utilities. The proof follows easily from Lagrangian duality and Euler's homogeneous function theorem [59, Section 6.2].

Lemma 3. *Let u be a utility function that is homogeneous of degree one and differentiable. Then, for any $(p,b) \in \mathbb{R}_+^{m+1}$, the optimal solutions to the systems* (1) *and* (4) *coincide.*

Nesterov and Shikhman [58] study Gale equilibria (equilibrium under Gale demand systems) as well as the more general concept of Fisher-Gale equilibria; they also give a tâtonnement type algorithm for finding such an equilibrium.

In the context of our work, in [38] we extend the auction algorithm for spending-restricted equilibria to Gale demand systems. This can be applied to approximating the Nash social welfare, as discussed in Introduction.

3 THE AUCTION ALGORITHM

Algorithm 1 describes the auction algorithm. It outputs a 4ε-approximate market equilibrium for an accuracy parameter $0 < \varepsilon < 0.25$ specified in the input. We use the notation e, $e^{(i)}$, D_i as in Definitions 1 and 2. We introduce some notation and formulate key invariants.

(a) We maintain a price vector p called *market prices*, initialized as $p_j = 1$ for all $j \in [m]$.[3] Prices may only increase, and remain integer powers of $(1+\varepsilon)$.

(b) No good is oversold, i.e., at most e_j amount of each good is sold, for $j = 1, 2, \ldots, m$. Also, the market price for each good j that is not fully sold is $p_j = 1$.

(c) The budget of agent i is $b_i = \langle p, e^{(i)}\rangle$. Every agent $i \in [n]$ maintains *individual prices* $p^{(i)} \in \mathbb{R}_+^m$ satisfying $p \le p^{(i)} \le (1+\varepsilon)p$. We let

$$L_i := \{j \in [m] : p_j^{(i)} < (1+\varepsilon)p_j\} \quad \text{and} \quad H_i := [m] \setminus L_i\,.$$

[3]Recall from Assumption 1 that if there exist market clearing prices that are strictly positive, we can also assume that these prices are at least 1. Even though there might be goods priced at 0 in an equilibrium, we can always find an ε-approximate market equilibrium where all prices are positive.

(d) Every agent $i \in [n]$ owns a bundle of goods $c^{(i)} \in \mathbb{R}^m_+$ that is dominated by a bundle $x^{(i)} \in D_i(p^{(i)}, b_i)$, i.e., an optimal bundle with respect to the individual prices $p^{(i)}$ and the budget b_i. We call $x^{(i)}$ the *desired bundle.*

(e) For the amount $c_j^{(i)}$ of good j, agent i pays p_j if $j \in L_i$ and $(1+\varepsilon)p_j$ if $j \in H_i$.[4] The *surplus* of agent i is

$$s_i := b_i - \sum_{j \in L_i} c_j^{(i)} p_j - (1+\varepsilon) \sum_{j \in H_i} c_j^{(i)} p_j \,.$$

Before giving an overview of the algorithm, we formulate the termination condition.

Lemma 4. *Assume that (a)–(e) hold as above. Then $s_i \geq 0$ for all $i \in [n]$. Moreover, if*

$$\sum_{i=1}^{n} s_i \leq 3\varepsilon \langle p, e \rangle$$

for the input bundle $e \in \mathbb{R}^m_+$, then the market prices p and allocations $c^{(i)}$, $i = 1, 2, \ldots, n$ form a 4ε-approximate market equilibrium.

Proof. Let $c^{(i)}$ be the bundle of goods agent i owns. By invariant (d), there exists a desired bundle $x^{(i)}$ dominating $c^{(i)}$. The bundle $x^{(i)}$ is affordable for i at prices $p^{(i)}$, and thus by invariants (d) and (e) the same bundle is affordable for i at prices p_j for $j \in L_i$ and $(1+\varepsilon)p_j$ for $j \in H_i$. It follows that i can afford $c^{(i)}$ at prices p_j for $j \in L_i$ and $(1+\varepsilon)p_j$ for $j \in H_i$. Hence, $s_i \geq 0$.

Condition (i) in Definition 2 is immediate from invariants (c) and (d), and condition (ii) follows from (b). It is left to verify condition (iii). We can write

$$\left\langle p, e - \sum_{i=1}^{n} c^{(i)} \right\rangle = \sum_{i=1}^{n} \left\langle p, e^{(i)} \right\rangle - \sum_{i=1}^{n} \left\langle p, c^{(i)} \right\rangle = \sum_{i=1}^{n} \left(b_i - \left\langle p, c^{(i)} \right\rangle \right)$$

$$= \sum_{i=1}^{n} \left(s_i + \varepsilon \sum_{j \in H_i} c_j^{(i)} p_j \right) \leq \sum_{i=1}^{n} s_i + \varepsilon \langle p, e \rangle \leq 4\varepsilon \langle p, e \rangle \;.$$

□

We now give an overview of the algorithm. The individual prices $p^{(i)}$ are updated by the key subroutine `FindNewPrices` that outputs prices and bundles as specified below. In Section 4, we provide implementations for different classes of demand systems.

> Subroutine `FindNewPrices`
>
> **Input:** Agent $i \in [n]$, market prices $p \in \mathbb{R}^m_+$, individual prices $p^{(i)} \in \mathbb{R}^m_+$ such that $p \leq p^{(i)} \leq (1+\varepsilon)p$, budget $b_i \in \mathbb{R}_+$, and bundle $c^{(i)} \in \mathbb{R}^m_+$.
>
> **Output:** Prices $\tilde{p} \in \mathbb{R}^m_+$ and bundle $y \in \mathbb{R}^m_+$ such that
> (A) $y \in D_i(\tilde{p}, b_i)$ and $y \geq c^{(i)}$, and
> (B) $p^{(i)} \leq \tilde{p} \leq (1+\varepsilon)p$, and $\tilde{p}_j = (1+\varepsilon)p_j$ whenever $y_j > (1+\varepsilon)\, c_j^{(i)}$.

The new individual prices will be set as $\tilde{p}$ and the new desired bundle as y. Property (B) requires that if agent i wants significantly more of good j than the current amount $c_j^{(i)}$, then they are willing to pay the higher price $(1+\varepsilon)p_j$.

[4]This is in contrast with [41] and the other previous auction algorithms where i may pay p_j for some amount of good j and $(1+\varepsilon)p_j$ for another amount.

The auction algorithm (Algorithm 1) considers the agents one-by-one in steps. A step gives an agent i a chance to spend her surplus money s_i to obtain more goods. If $s_i > 0$, agent i calls `FindNewPrices`$(i, p^{(i)}, p, b_i, c^{(i)})$ to obtain new individual prices $\tilde{p}$ and desired bundle y. At the end of their step, they update $p^{(i)} = \tilde{p}$.

Given $\tilde{p}$ and y, agent i considers all goods with $\tilde{p}_j = (1+\varepsilon)p_j$ one-by-one, and tries to purchase $y_j - c_j^{(i)}$ amount using the `Outbid` procedure. First, if $\sum_{i=1}^n c_j^{(i)} < e_j$, i.e., if there is any unsold amount of good j, they purchase from such amounts. If they still want more, they will outbid other agents who have been paying the lower price p_j for this good, by offering the higher price $(1+\varepsilon)p_j$. Goods with $\tilde{p}_j < (1+\varepsilon)p_j$ are ignored: no additional amount of these goods is purchased.

If after the calls to `Outbid`, a good j is only being sold at the higher price $(1+\varepsilon)p_j$, then we call the `RaisePrice` procedure to increase the market price from p_j to $(1+\varepsilon)p_j$, and update the budgets and surpluses accordingly. The algorithm terminates once the total surplus of the agents is at most $3\varepsilon\langle p, e\rangle$; according to Lemma 4, the current prices and allocations form a 4ε-approximate equilibrium.

We now formulate the main running time statement. This depends on the running time T_F of the subroutine `FindNewPrices`. We assume that $T_F = \Omega(m)$, since the output needs to return an m-dimensional vector of goods.

We also use an upper bound on $p_{\max}/p_{\min}$—the ratio of the largest and smallest nonzero prices at some ε-equilibrium. An upper bound on $p_{\max}/p_{\min}$ may be obtained for the specific demand systems, e.g., for demand systems arising from linear utilities [41]. Alternatively, one can follow the approach of Codenotti, McCune, and Varadarajan [21, 23] by adding a dummy agent with a Cobb–Douglas demand system and an initial endowment of a small fraction of all goods. In the presence of such an agent, we can obtain a strong bound on $p_{\max}/p_{\min}$, at the expense of obtaining a slightly worse approximation guarantee. We describe the construction in Appendix B.

Note that it is the ratio $p_{\max}/p_{\min}$ rather than the actual values of $p_{\max}$ and $p_{\min}$ that matter: by Assumption 1, for (approximate-)equilibrium prices p, αp also gives (approximate-)equilibrium prices with the same allocation, for any $\alpha > 0$. In our algorithm, the minimum price will remain 1 throughout, see Lemma 6.

Theorem 1. *Assume all agents have WGS demand systems. Let T_F be an upper bound on the running time of the subroutine* `FindNewPrices` *and suppose that $T_F = \Omega(m)$. Algorithm 1 finds a 4ε-approximate market equilibrium in time $O\left(\frac{nmT_F}{\varepsilon^2} \cdot \log\left(\frac{p_{\max}}{p_{\min}}\right)\right)$.*

One particular implementation of `FindNewPrices` in Section 4.1 is given for bounded elasticities; see Lemma 9 for the running time-bound. Recall the elasticity bound f from Definition 6.

Theorem 2. *If all agents have WGS demand systems with elasticity at least $-f$ for some $f > 0$, then an ε-approximate equilibrium can be computed in time $O\left(\frac{nm^2 f \cdot T_D}{\varepsilon^2} \cdot \log\left(\frac{p_{\max}}{p_{\min}}\right)\right)$, where T_D is the time needed for one call to the demand oracle.*

We give an overview of the running times of the previous auction algorithms in Section A.

3.1 Description of the algorithm

We now give a more detailed overview of the algorithm. Recall the notation and invariants (a)–(e) at the start of this section.

For each good $j = 1, 2, \ldots, m$, we partition the total amount as $e_j = w_j + l_j + h_j$ according to the price it is sold at:

- amount w_j is the unsold part of the good,

Algorithm 1: Auction algorithm

Input: Demand systems D_i, and the endowment vectors $e^{(i)}$, and $\varepsilon \in (0, 0.25)$.
Output: A 4ε-approximate market equilibrium.

Initialize
  for $j \in [m]$ **do** $e_j \leftarrow \sum_{i=1}^{n} e_j^{(i)}$; $p_j \leftarrow 1$; $w_j \leftarrow e_j$; $l_j \leftarrow 0$
  for $i \in [n]$ **do**
    $b_i \leftarrow \sum_{j=1}^{n} e_j^{(i)}$; $s_i \leftarrow b_i$
    for $j \in [m]$ **do** $p_j^{(i)} \leftarrow 1$; $c_j^{(i)} \leftarrow 0$
  end
while $\sum_{i=1}^{n} s_i > 3\varepsilon \langle p, e \rangle$ **do**
  Select next agent $i \in [n]$ with $s_i > 0$. // Step for agent i.
  $(\tilde{p}, y) \leftarrow$ `FindNewPrices`$(i, p, p^{(i)}, b_i, c^{(i)})$
  for $j = 1$ **to** m **do**
    if $p_j^{(i)} < (1+\varepsilon)p_j$ *and* $\tilde{p}_j = (1+\varepsilon)p_j$ **then** // Case 1
      $s_i \leftarrow s_i - c_j^{(i)} \cdot \varepsilon p_j$; $l_j \leftarrow l_j - c_j^{(i)}$ // i pays $(1+\varepsilon)p_j$ instead of p_j.
      `Outbid`$(i, j, y_j - c_j^{(i)})$
    else if $p_j^{(i)} = (1+\varepsilon)p_j$ *and* $\tilde{p}_j = (1+\varepsilon)p_j$ **then** // Case 2
      `Outbid`$(i, j, y_j - c_j^{(i)})$
    end
    // Skip the goods with $p_j^{(i)} < (1+\varepsilon)p_j$ and $\tilde{p}_j < (1+\varepsilon)p_j$. // Case 3
    if $w_j + l_j = 0$ **then** `RaisePrice`(j)
  end
  $p^{(i)} \leftarrow \tilde{p}$
end
return p, $\{p^{(i)}\}_{i\in[n]}$ and $\{c^{(i)}\}_{i\in[n]}$

- amount l_j is sold at the lower price p_j, and
- amount h_j is sold at the higher price $(1+\varepsilon)p_j$.

We only explicitly maintain w_j and l_j in the algorithm. Recall from (b) that $w_j \geq 0$ and $p_j = 1$ as long as $w_j > 0$. We further maintain $w_j + l_j > 0$ at the beginning of every step, i.e., there is always a part of the good that is unsold or owned by an agent at the lower price.

The Outbid subroutine. Procedure `Outbid`(i, j, γ), controls how the ownership of goods may change. This is called when agent i would like to purchase an additional amount γ of good j.

If $w_j > 0$, there is some unsold amount of the good, then agent i starts by purchasing $\min\{w_j, \gamma\}$ of this amount. Recall that $p_j = 1$ at this point is due to invariant (b).

If agent j still requires more of good j, we consider agents k one-by-one who are paying the lower price p_j for good j, i.e., $j \in L_k$. Agent i may take over some of this amount by offering a higher price $(1+\varepsilon)p_j$.

The RaisePrice subroutine. Procedure `RaisePrice`(j) is called when $w_j + l_j = 0$ for a good j, i.e., it is only sold at the higher price $(1+\varepsilon)p_j$. In this case, we increase the market price to $(1+\varepsilon)p_j$,

Procedure Outbid(i, j, γ)

Input: Agent $i \in [n]$, good $j \in [m]$, amount $\gamma > 0$.

$z \leftarrow \gamma$
if $w_j > 0$ **then** // a part of j is unsold
  $\mu \leftarrow \min\{w_j, z\}$
  $w_j \leftarrow w_j - \mu$
  $c_j^{(i)} \leftarrow c_j^{(i)} + \mu$
  $s_i \leftarrow s_i - (1+\varepsilon)\mu$ // here $p_j = 1$
  $z \leftarrow z - \mu$
end
while $z > 0$ *and* $l_j > 0$ **do**
  Let $k \in [n]$ be such that $c_j^{(k)} > 0$ and $j \in L_k$
  $\mu \leftarrow \min\{c_j^{(k)}, z\}$
  $l_j \leftarrow l_j - \mu$
  $c_j^{(k)} \leftarrow c_j^{(k)} - \mu$; $c_j^{(i)} \leftarrow c_j^{(i)} + \mu$ // i outbids k
  $s_k \leftarrow s_k + \mu p_j$; $s_i \leftarrow s_i - (1+\varepsilon)\mu p_j$
  $z \leftarrow z - \mu$
end

Procedure RaisePrice(j)

Input: Good $j \in [m]$.

for $k \in [n]$ **do**
  $b_k \leftarrow b_k + \varepsilon p_j e_j^{(k)}$
  $s_k \leftarrow s_k + \varepsilon p_j e_j^{(k)}$
  $p_j^{(k)} \leftarrow (1+\varepsilon)p_j$
end
$p_j \leftarrow (1+\varepsilon)p_j$; $l_j \leftarrow e_j$

set all individual prices $p_j^{(k)}$ to this value, and update the budgets and surpluses of all agents whose initial endowment contains good j. We also set $l_j = e_j$.

Steps. The algorithm terminates as soon as the total surplus is at most $3\varepsilon \langle p, e \rangle$. We consider agents i with $s_i > 0$ one-by-one. By invariant (d), the agent is buying a bundle $c^{(i)} \le x^{(i)}$ for some $x^{(i)} \in D_i(p^{(i)}, b_i)$. The subroutine `FindNewPrices`$(i, p^{(i)}, p, b_i, c^{(i)})$ delivers new prices $\tilde{p}$ and a bundle y satisfying Conditions (A) and (B) described above.

Conditions (A) asserts that the current bundle $c^{(i)}$ of agent i is still dominated by a desired bundle y at the increased prices $\tilde{p}$. Condition (B) guarantees that $\tilde{p} \ge p^{(i)}$, and whenever an agent wants to buy more of some good than they already own at least by a factor $(1+\varepsilon)$, then they are willing to pay the higher price $(1+\varepsilon)p_j$ for it. (They might already be paying the increased price to start with if $p_j^{(i)} = (1+\varepsilon)p_j$. In this case $\tilde{p}_j = (1+\varepsilon)p_j = p_j^{(i)}$.)

The above properties suggest the following update rules for each good $j \in [m]$.

Case 1. $p_j^{(i)} < (1+\varepsilon)p_j$ *and* $\tilde{p}_j = (1+\varepsilon)p_j$. The good j was in L_i and needs to be moved to H_i, i.e., agent i used to pay p_j but now is willing to pay the higher price for j. We charge agent i the price $(1+\varepsilon)p_j$ for the amount $c_j^{(i)}$ they already own instead of p_j. Additionally, agent i outbids on good j the amount $\min\{y_j - c_j^{(i)}, w_j + l_j\}$.

Case 2. $p_j^{(i)} = (1+\varepsilon)p_j$ *and* $\tilde{p}_j = (1+\varepsilon)p_j$. The good j was in H_i and stays in H_i, i.e., agent i continues to pay the higher price. The agent i keeps the amount $c_j^{(i)}$ of good j that they already had and outbids for as much as they can from the other agents, i.e., the amount $\min\{y_j - c_j^{(i)}, w_j + l_j\}$.

Case 3. $p_j^{(i)} < (1+\varepsilon)p_j$ *and* $\tilde{p}_j < (1+\varepsilon)p_j$. The good j remains in L_i, i.e., agent i continues to pay the lower price. By (B), we must have $c_j^{(i)} \le y_j \le (1+\varepsilon)c_j^{(i)}$; the agent will not seek to buy more of these goods.

If the above updates result in $w_j + l_j = 0$ for good j, then we call `RaisePrice(`j`)` to increase the market price. Once all of the goods have been considered we set $p^{(i)} = \tilde{p}$ and update $c^{(i)}$ as the current allocation.

Rounds. In the analysis, it will be useful to organize the steps into rounds as in [41]. A *round* consists of all agents making a step once (or being skipped if $s_i = 0$).

3.2 Analysis

We start with a high-level overview of the analysis. Lemma 5 verifies that all invariants (a)–(e) are maintained. Lemma 6 shows that the minimum price remains $p_j = 1$ throughout. Note that—in accordance with invariant (b)—it suffices to show that $w_j > 0$ for some good j holds at any point. In other words, the algorithm terminates if all goods are fully sold.

Lemma 7 is the key argument in the running time bound: it shows that the number of rounds between two calls to `RaisePrice` is bounded by $2/\varepsilon$. The idea is that as long as the market prices do not increase, the total budget ($\sum_{i=1}^{n} b_i$) remains the same, while the total money spent on the goods is increasing due to outbidding. In every outbid, an agent pays more by a factor $(1+\varepsilon)$ for the goods they purchase. Thus, in n consecutive steps the total surplus decreases by approximately a factor $(1+\varepsilon)$. This either leads to a market price increase or reaches an approximate equilibrium.

Lemma 5. *If all agents have WGS demand systems, then the invariants (a)-(e) hold after every step.*

Proof. **(a)** This is immediate.

- **(b)** The algorithm maintains $e_j = w_j + l_j + h_j$ for all goods $j = 1, 2, \ldots, m$. Further, $w_j, l_j, h_j \ge 0$. The amount sold is $l_j + h_j \le e_j$, hence, no good is oversold. The amount w_j is monotone decreasing throughout. This is guaranteed by Property (A) of the procedure `FindNewPrices`, and the fact that $c_j^{(i)}$ may only decrease if another $c_j^{(k)}$ increases by the same amount. Thus, if $w_j = 0$ at any point of the algorithm, good j remains fully sold in all remaining steps. Note that the market price p_j is first increased from the initial value $p_j = 1$ when $w_j = l_j = 0$.
- **(c)** The budgets are updated in `RaisePrice`, adding $\varepsilon p_j e_j^{(i)}$ to b_i when the price p_j increases to $(1+\varepsilon)p_j$. Thus, we maintain $b_i = \langle p, e^{(i)} \rangle$. The bounds $p \le p^{(i)} \le (1+\varepsilon)p$ are immediate from condition (B) in `FindNewPrices` and in the procedure `RaisePrice`.
- **(d)** Suppose these properties hold for every agent before a step of agent i. The requirements (A) and (B) guarantee that $c^{(i)}$ is dominated by a bundle $x^{(i)} \in D_i(p^{(i)}, b_i)$ and prices satisfy $p \le p^{(i)} \le (1+\varepsilon)p$, for each agent i. Moreover, if the budget b_i is increased in line 2, the invariant remains true by the WGS property.

Now, consider an agent k different from i. In the step, k could lose a part of a good only through the outbid and hence $c^{(k)}$ does not increase. As long as the prices $p^{(k)}$ do not change, (d) holds trivially. The only time $p^{(k)}$ can change is the price increase step in line 4, namely, if p_j increases to $(1+\varepsilon)p_j$, it forces $p_j^{(k)} = (1+\varepsilon)p_j$. Note that the price increase only happens once $l_j = 0$. Assume we had $p_j^{(k)} < (1+\varepsilon)p_j$ before the price increase, that is, agent k was buying good j at the lower price p_j. By $l_j = 0$ and invariant (e), it follows that $c_j^{(k)} = 0$ at this point. As the budgets may only increase, the WGS property implies that after increasing $p_j^{(k)}$, the bundle $c^{(k)}$ will still be dominated by an optimal bundle.

(e) It is straightforward to check that the form of the surplus is maintained.

□

Lemma 6. *At the beginning of every step,* $\min\{p_j : j \in G\} = 1$.

Proof. We show that a good j with $w_j > 0$ exists in every step. The statement then follows by (b). For a contradiction, suppose we reach a point where $w_j = 0$ for all $j = 1, 2, \ldots, m$, and consider the first time this happens. At this point all of the goods are fully sold, i.e., $\sum_{i=1}^n c^{(i)} = e$. Consider the total surplus at this point. We have

$$\sum_{i=1}^n s_i = \sum_{i=1}^n \left(b_i - \sum_{j \in L_i} p_j c_j^{(i)} - \sum_{j \in H_i} (1+\varepsilon) p_j c_j^{(i)} \right) \le \sum_{i=1}^n \left(b_i - \left\langle p, c^{(i)} \right\rangle \right)$$
$$= \sum_{i=1}^n b_i - \left\langle p, \sum_{i=1}^n c^{(i)} \right\rangle = \sum_{i=1}^n b_i - \langle p, e \rangle = 0\,.$$

This contradicts the assumption $\sum_{i=1}^n s_i > 3\varepsilon \langle p, e \rangle$ at the beginning of every step. □

Lemma 7. *The number of rounds between any two consecutive calls to* `RaisePrice` *is at most* $2/\varepsilon$.

Proof. Let p be the market prices after a call to `RaisePrice`, and consider a sequence of steps with the same market prices; consequently, the budget of every agent remains the same. Consider a step of an agent i during this sequence. If i buys $w_j + l_j$ of a good j, then `RaisePrice` is called, and the sequence finishes. Thus, we can assume that during this sequence, every agent i gets the amount of each good they desire.

Let φ denote the total amount of money spent at a certain point of this sequence of steps that is spent by the agents on higher price goods. That is,

$$\varphi = (1+\varepsilon) \sum_{i=1}^n \sum_{j \in H_i} c_j^{(i)} p_j.$$

Claim 1. *Let s_i denote the surplus of agent i at the beginning of their step. Then the value of φ increases at least by $(1+\varepsilon)^2 s_i - 2.25\varepsilon b_i$ during agent i's step.*

Proof: Recall Cases 1-3 in the description of the step. Let T_k be the set of goods that fall into case k, that is, $T_1 \cup T_2 \cup T_3 = [m]$.

- If $j \in T_1$, then $(1+\varepsilon)p_j y_j$ will be added to φ in the `Outbid` subroutine: In this case, the agent also outbids itself, moving good j from L_i to H_i.
- If $j \in T_2$, then $(1+\varepsilon)p_j(y_j - c_j^{(i)})$ will be added to φ in the `Outbid` subroutine.

- If $j \in T_3$, then we do not increase φ. Nevertheless, (B) guarantees that $\tilde{p}_j(y_j - c_j^{(i)}) \le \varepsilon \tilde{p}_j c_j^{(i)}$. Consequently,

$$\sum_{j \in T_3} \tilde{p}_j(y_j - c_j^{(i)}) \le \varepsilon \left\langle \tilde{p}, c^{(i)} \right\rangle . \tag{5}$$

Also note that $\tilde{p}_j = (1+\varepsilon)p_j$ if $j \in T_1 \cup T_2$. Assumption 2 on non-satiation guarantees that $\langle \tilde{p}, y \rangle = b_i$. Let $\Delta\varphi$ denote the increment in φ; this can be lower bounded as

$$\begin{aligned}\Delta\varphi &= \sum_{j \in T_1} \tilde{p}_j y_j + \sum_{j \in T_2} \tilde{p}_j(y_j - c_j^{(i)}) \\ &= \langle \tilde{p}, y \rangle - \sum_{j \in T_3} \tilde{p}_j y_j - \sum_{j \in T_2} \tilde{p}_j c_j^{(i)} \\ &\ge b_i - \sum_{j \in T_3} \tilde{p}_j(y_j - c_j^{(i)}) - \left\langle \tilde{p}, c^{(i)} \right\rangle \\ &\ge b_i - (1+\varepsilon) \left\langle \tilde{p}, c^{(i)} \right\rangle ,\end{aligned}$$

using (5). The money spent by the agent at the beginning of the step is $b_i - s_i$. Good j is purchased at price at least p_j according to (e), and $\tilde{p}_j \le (1+\varepsilon)p_j$. Consequently, $\left\langle \tilde{p}, c^{(i)} \right\rangle \le (1+\varepsilon)(b_i - s_i)$. With the above inequality, we obtain

$$\Delta\varphi \ge b_i - (1+\varepsilon)^2(b_i - s_i) \ge (1+\varepsilon)^2 s_i - (2\varepsilon + \varepsilon^2) b_i \ge (1+\varepsilon)^2 s_i - 2.25\varepsilon b_i,$$

as $\varepsilon < 0.25$. This completes the proof. ■

As long as $\sum_{i=1}^n s_i > 3\varepsilon \langle p, e \rangle$, the claim guarantees that φ increases in every round by at least

$$3(1+\varepsilon)^2 \varepsilon \langle p, e \rangle - 2.25\varepsilon \sum_{i=1}^n b_i = 0.75\varepsilon \langle p, e \rangle .$$

Since $\varphi \le (1+\varepsilon) \langle p, e \rangle$, the number of rounds until the next call the `RaisePrice` is bounded by $2/\varepsilon$. □

We need one more lemma to bound the total running time of `Outbid`.

Lemma 8. *Between two calls to* `RaisePrice`*, for every agent $k \in [n]$ and good $j \in [n]$, there can be only one occasion that during a call to* `Outbid`*(i, j, γ) for some $i \in [n]$ and $\gamma > 0$, $c_j^{(k)}$ is set to 0 in line 13.*

Proof. Assume $c_j^{(k)}$ was set to 0 in a certain call to `Outbid`. Before it can be set to 0 again, agent k must have obtained a positive amount of good j. This could only happen in a call to `Outbid`(k, j, γ) for some $\gamma > 0$. However, until the next call to `RaisePrice`, agent k may only buy good j at the higher price $(1+\varepsilon)p_j$, and henceforth $j \in H_k$. Thus, agent k cannot be selected again in line 10 at a call to `Outbid`(i, j, γ). □

Proof of Theorem 1. Lemma 4 shows that at termination, the algorithm returns a 4ε-market equilibrium.

We first bound the total running time between two calls to `RaisePrice`. By Lemma 7, there are at most $2/\varepsilon$ rounds. Every round comprises n steps, and every step calls the procedure `FindNew-Prices` exactly once. Therefore, the time taken by `FindNewPrices` during this sequence of steps is $O(nT_F/\varepsilon)$.

The total number of calls to `Outbid` is m in each step, totaling to $O(nm/\varepsilon)$. We bound the number of repeats in the 'while' loop (lines 9–14) in all calls to `Outbid` between two calls to `RaisePrice`. In a call to `Outbid`(i, j, γ), in all but the final call to the 'while' loop, we set $c_j^{(k)} = 0$ for some agent

k. By Lemma 8, the total number of these events is at most $O(nm)$. Hence, the number of repeats in the 'while' loop between two calls to `RaisePrice` is $O(nm/\varepsilon + nm) = O(nm/\varepsilon)$. Each repeat takes $O(1)$ time.

From the above, the total time of the `Outbid` calls is $O(nm/\varepsilon)$ between two calls to `RaisePrice`. A call to `RaisePrice` takes $O(nm)$ time. Consequently, the total time of such a sequence of steps is $O(nT_F/\varepsilon + nm/\varepsilon) = O(nT_F/\varepsilon)$, using the assumption that $T_F = \Omega(m)$.

By Lemma 6, the minimum price remains at most 1 throughout and therefore $p_{\max}$ is at most $p_{\max}/p_{\min}$. Consequently, `RaisePrice` can be called at most $O(m \log_{1+\varepsilon}(\frac{p_{\max}}{p_{\min}})) = O(\frac{m}{\varepsilon}\log(\frac{p_{\max}}{p_{\min}}))$ times. The claimed running time bound follows. □

4 IMPLEMENTING THE PRICE UPDATE SUBROUTINE

In this section, we present different approaches to implement `FindNewPrices`$(i, p, p^{(i)}, b_i, c^{(i)})$. Recall that the output prices $\tilde{p} \in \mathbb{R}^m_+$ and allocations $y \in \mathbb{R}^m_+$ must satisfy the following two requirements:

(A) $y \in D_i(\tilde{p}, b_i)$ and $y \geq c^{(i)}$, and
(B) $p^{(i)} \leq \tilde{p} \leq (1+\varepsilon)p$, and $\tilde{p}_j = (1+\varepsilon)p_j$ whenever $y_j > (1+\varepsilon)\, c^{(i)}_j$.

First, in Section 4.1 we consider the setting of bounded elasticities. Recall from Lemma 1 that this includes Cobb–Douglas utilities and CES utilities with parameter $\sigma > 1$. Further, according to Lemma 2, we can use it for convex combinations of demand systems with bounded elasticities, even if they are not given in the explicit form (1). The algorithm is a simple price increment procedure, making repeated calls to the demand oracle. As linear utilities do not have bounded elasticities, in Section 4.2, we give a simple direct algorithm for linear demand systems. Finally, in Section 4.3, we implement `FindNewPrices` for Gale demand systems. We obtain the new prices as the optimal Lagrangian multipliers of a convex program.

Note that for many utility functions, such as Cobb–Douglas or CES, we can use either of the methods in Section 4.1 or Section 4.3. The running time in Section 4.1 depends linearly on the elasticity parameter f and makes several calls to the demand oracle. Still, it could be faster than solving a convex program, e.g., if the demand oracle is given by an explicit formula.

It is possible to find further direct approaches for particular demand systems, similar to the approach in Section 4.2 for linear demand systems. For example, it is easy to devise an $O(m)$ time procedure for Cobb–Douglas demand systems, exploiting the fact that the optimal bundle allocates $\alpha^{(i)}_j b_i$ money for good j. Consequently, each price can be set independently of the others.

We note that even though the above cases cover all standard examples of WGS systems, we do not have a general implementation for demand systems in the form (1).

4.1 Demand systems with bounded elasticities

Let us assume that the demand system D_i has elasticity of at least $-f$ for some $f > 0$. The subroutine `BE-FindNewPrices`$_f(i, p, p^{(i)}, b_i, c^{(i)})$ (Algorithm 2) is a simple price increment procedure. First, we obtain $y \in D_i(p^{(i)}, b_i)$ from the demand oracle with $y \geq c^{(i)}$. This is possible due to invariant (d), which guarantees that $c^{(i)} \leq x^{(i)}$ for some $x^{(i)} \leq D_i(p^{(i)}, b_i)$. Thus, $y = x^{(i)}$ is itself a suitable choice. Then, we iterate the following step. As long as (B) is violated for a good j, we increase its price by a factor $(1+\varepsilon)^{1/f}$ until it reaches the upper bound $(1+\varepsilon)p_j$.

Lemma 9. *Assume the demand system D_i has elasticity of at least $-f$ for some $f > 0$. Algorithm 2 terminates with $\tilde{p}$ and y satisfying (A) and (B) in time $O(mf \cdot T_D)$, where T_D is the time for a call to the demand oracle.*

We assume that $T_D = \Omega(m)$, since the demand oracle needs to output an m-dimensional vector.

Algorithm 2: BE-FindNewPrices$_f(i, p, p^{(i)}, b_i, c^{(i)})$

Input: Agent $i \in [n]$, market prices $p \in \mathbb{R}^m_+$, individual prices $p^{(i)} \in \mathbb{R}^m_+$ such that $p \leq p^{(i)} \leq (1+\varepsilon)p$, budget $b_i \in \mathbb{R}_+$, and bundle $c^{(i)} \in \mathbb{R}^m_+$.

Output: Prices $\tilde{p}$ and bundle y.

Initialization: $\tilde{p} \leftarrow p^{(i)}$

Obtain $y \in D_i(\tilde{p}, b_i)$ from the demand oracle such that $y \geq c^{(i)}$

while $\exists j : \tilde{p}_j < (1+\varepsilon)p_j$ *and* $y_j > (1+\varepsilon)c^{(i)}_j$ **do**

  $\tilde{p}_j \leftarrow \min\{(1+\varepsilon)^{1/f}\tilde{p}_j, (1+\varepsilon)p_j\}$

  Obtain $y' \in D_i(\tilde{p}, b_i)$ from the demand oracle such that $y'_k \geq y_k$ for $k \neq j$

  $y \leftarrow y'$

end

return $(\tilde{p}, y)$

Proof. The bound on the number of iterations is clear: since we have $p \leq \tilde{p} \leq (1+\varepsilon)p$ throughout, the price of every good can increase at most f times. Condition (A) is satisfied due to the WGS property and the bound on the demand elasticity: when increasing $\tilde{p}_j$, the demand y_k for $k \neq j$ is non-decreasing as guaranteed by the demand oracle. Further, y_j may decrease only by a factor $(1+\varepsilon)$, and since we had $y_j > (1+\varepsilon)c^{(i)}_j$ before the price update, we still have $y_j > c^{(i)}_j$ after the price update. Condition (B) is satisfied at termination since the while loop keeps running as long as it is violated. Checking the while condition each time requires $O(m)$ time; however, this will be dominated by the time T_D according to the comment on $T_D = \Omega(m)$ above. □

4.2 Linear demand systems

We now give a simple direct implementation of FindNewPrices for linear demand systems.

Lemma 10. FindNewPrices *can be implemented in $O(m)$ for a linear demand system corresponding to the utility function $u(x) = \langle v, x\rangle$.*

Proof. Recall that for linear utilities $y \in D_i(\tilde{p}, b)$, $y_j > 0$ if and only if $j \in \arg\max_k v_k/p_k$, called *maximum bang-per-buck goods (MBB)*. We initialize $\tilde{p} = p^{(i)}$, and let $S \subseteq [m]$ denote the set of MBB goods. Thus, $y_j = 0$ for all $j \notin S$. We start increasing the prices of all goods $j \in S$ at the same rate α. Once a good outside S becomes MBB, we include it in the set S and also start raising its price. We terminate when the budget is exhausted or when the price $\tilde{p}_k$ for a good $k \in S$ reaches the upper bound $(1+\varepsilon)p_k$. In the former case, we return the bundle $y_j = c^{(i)}_j, \forall j$. In the latter case, we return the bundle $y_j = c^{(i)}_j$ if $j \neq k$, and set $y_k = (b_i - \sum_{j\in S; j\neq k} \tilde{p}_j c^{(i)}_j)/\tilde{p}_k$; clearly, $y_k \geq c^{(i)}_k$. These prices and allocations satisfy (A) and (B); in fact, we obtain (B) in the stronger form that $\tilde{p}_j = (1+\varepsilon)p_j$ whenever $y_j > c^{(i)}_j$. We need to add a good to S at most m times, and thus we can implement the procedure in $O(m)$ time. □

4.3 Gale demand systems

We now show that the subroutine FindNewPrices can be implemented for Gale demand systems via convex programming. According to Lemma 3, this result is also applicable to demand systems given in the form (1) for utility functions that are homogeneous of degree one, in which case the optimal solutions to (1) and (4) coincide.

Suppose the utility function $u : \mathbb{R}^m_+ \to \mathbb{R}_+$ is strictly concave and differentiable. Strict concavity implies that the demand system is simple: $|G^u(p, b)| = 1$ for all $(p, b) \in \mathbb{R}^m_+$.

We implement a stronger and more general form of `FindNewPrices`, with an arbitrary vector $q \in \mathbb{R}^m_+$, $q \geq p$ in place of $(1+\varepsilon)p$.

Assume we are given $b \in \mathbb{R}_+$, $p, q, c \in \mathbb{R}^m_+$ such that $p \leq q$, and moreover assume that $c \leq x$ for some $x \in G^u(p, b)$. The goal is to find $\tilde{p}$ and y such that

(A') $y \geq c$ where $y \in G^u(\tilde{p}, b)$, and
(B') $p \leq \tilde{p} \leq q$ and $\tilde{p}_j = q_j$ whenever $y_j > c_j$.

In the following convex program, the agent is allowed to buy a good j at two prices: amount y'_j at price p_j and amount y''_j at price q_j; the amount at the lower price p_j is capped at c_j.

$$\begin{aligned} \max\ & b \ln u(y) - \langle p, y' \rangle - \langle q, y'' \rangle \\ & y = y' + y'' \\ & y' \leq c \\ & y', y'' \geq 0\,. \end{aligned} \tag{6}$$

We show that the optimal solution to this program, along with the prices obtained from the KKT conditions satisfy (A') and (B').

Since all constraints are linear, strong duality holds. Let $y^* = y' + y''$ be an optimal solution of (6). By the KKT conditions, there exists $\alpha \in \mathbb{R}^m_+$ such that for any $j \in [m]$,

(i) $b \cdot \frac{\partial_j u(y^*)}{u(y^*)} \leq \min\{\alpha_j + p_j, q_j\}$,
(ii) $b \cdot \frac{\partial_j u(y^*)}{u(y^*)} = \alpha_j + p_j$ whenever $y'_j > 0$,
(iii) $b \cdot \frac{\partial_j u(y^*)}{u(y^*)} = q_j$ whenever $y''_j > 0$, and
(iv) $y'_j = c_j$ whenever $\alpha_j > 0$.

Note that in an optimal solution, we must have $y'_j > 0$ whenever $c_j > 0$ and $y^*_j > 0$. We define the prices $\tilde{p}_j$ as

$$\tilde{p}_j := \begin{cases} q_j & \text{if } c_j = 0 \text{ and } y^*_j > 0\,, \\ \alpha_j + p_j & \text{otherwise}\,. \end{cases}$$

Lemma 11. *The allocations y^* and prices $\tilde{p}$ satisfy (A') and (B').*

Proof. Since all constraints are linear, strong duality holds for (4) as well as for (6). We start with (B'). Let $j \in [m]$. If $c_j = 0$ and $y^*_j > 0$, then $\tilde{p}_j = q_j$ and thus (B') holds by definition.

Suppose $c_j > 0$. In case $y''_j > 0$, (ii) and (iii) imply that $\tilde{p}_j = \alpha_j + p_j = q_j$. This holds whenever $y^*_j > c_j$. It is left to show that $p_j \leq \tilde{p}_j \leq q_j$ when $y^*_j \leq c_j$. If $y^*_j > 0$, this follows from (i) and (ii). If $y^*_j = y'_j = 0$, by $c_j > 0$ and (iv) we have $\alpha_j = 0$ and thus $\tilde{p}_j = p_j$.

For (A'), we first show $y^* \in G^u(\tilde{p}, b)$. By the KKT conditions for (4), we have $y^* \in G^u(\tilde{p}, b)$ if and only if for all $j \in [m]$:

(G1) $\frac{b \partial_j u(y^*)}{u(y^*)} \leq \tilde{p}_j$, and
(G2) $\frac{b \partial_j u(y^*)}{u(y^*)} = \tilde{p}_j$ whenever $y^*_j > 0$.

Let $j \in [m]$. The condition $\frac{b \partial_j u(y^*)}{u(y^*)} \leq \tilde{p}_j$ follows by the definition of $\tilde{p}$ and by (i). Suppose $y^*_j = y'_j + y''_j > 0$. If $c_j = 0$, then $y''_j > 0$ and $\tilde{p}_j = q_j$; (G2) follows by (iii). If $c_j > 0$, then we must have $y'_j > 0$ and $\tilde{p}_j = p_j + \alpha_j$. Thus, (G2) follows from (ii).

It remains to show that $y^* \geq c$. We prove by contradiction: assume that $y^*_j < c_j$ for some good j, which implies $c_j > 0$. This implies $\alpha_j = 0$ by (iv), yielding $\tilde{p}_j = p_j$. By the strict concavity assumption, y^* is the unique optimal bundle in $G^u(\tilde{p}, b)$. Using the WGS property for (p, b) and $(\tilde{p}, b)$ we have $y^*_j \geq x_j$ since $p_j = \tilde{p}_j$. We obtain a contradiction to $y^*_j < c_j \leq x_j$. ◻

A RUNNING TIMES OF PREVIOUS AUCTION ALGORITHMS

We review the running time bounds given in previous auction algorithms and compare them to our bounds. We let $\mathbb{1}$ denote the n dimensional vector with all entries 1.

Linear utility functions, Garg and Kapoor [41]. The paper includes two algorithms. The running times are $O\left(\frac{nm}{\varepsilon^2}\cdot \log\left(\frac{p_{\max}\cdot\langle \mathbb{1},e\rangle}{\varepsilon\cdot p_{\min}\cdot e_{\min}}\right)\cdot \log\left(\frac{p_{\max}}{p_{\min}}\right)\right)$ and $O\left(\frac{nm}{\varepsilon}(n+m)\log\left(\frac{p_{\max}}{p_{\min}}\right)\right)$, respectively. The running time in Theorem 1, with the bound $T_F = O(m)$ for linear utilities from Lemma 10, gives an additional factor m when compared to the first bound, while removing the first log term. The additional factor is due to our global update step: due to the more general, nonseparable nature of our framework, we consider all goods when updating an agent, while the paper [41] considers only one good for an update. We note that we are using a weaker notion of equilibrium in our result.

Separable WGS utilities, Garg, Kapoor, and Vazirani [44]. The running time bound is presented only for the Fisher market case, given as $O\left(\frac{nm}{\varepsilon}\log\frac{1}{\varepsilon}\log\frac{v_{\max}\cdot\langle \mathbb{1},b\rangle}{b_{\min}v_{\min}}\log m\right)$. Here, $v_{\max} := \max_i v_{i\max}$ and $v_{\min} := \min_i v_{i\min}$ are upper and lower bounds on the slopes of the utility functions, $b_{\min}$ is the smallest budget and v is the total utility an agent would get from owning the full amount of all goods. An issue with such a bound is that the value $\frac{v_{\max}}{v_{\min}}$ is not scale invariant. Namely, the equilibrium in a Fisher market remains the same even if each agent i multiplies their utility function by a positive constant α_i; but this changes the value $\frac{v_{\max}}{v_{\min}}$ arbitrarily. It is mentioned that the result could be extended to exchange markets, similarly as in [41], but no details or running time estimation are provided.

Uniformly separable WGS utilities, Garg and Kapoor [43]. The paper gives essentially the same bound as in the case of separable WGS; the analysis is limited and mainly refers to the analysis of the auction algorithm for separable WGS utilities [44]. A problematic issue is that the main motivation for the paper is to give bounds for CES and Cobb-Douglas utilities, but $v_{\max} = \infty$ for these particular utilities.

B ADDING A DUMMY AGENT TO BOUND THE PRICES

Using the construction in the papers [21, 23], we present a general technique for modifying markets in order to bound $p_{\max}/p_{\min}$ in Theorem 1. Given an exchange market M with agents $A = [n]$ and goods $G = [m]$, we transform it to another market $\hat{M}$ with $n+1$ agents as follows. Assume $\varepsilon > 0$ is chosen such that $\varepsilon(1+\varepsilon)m < \frac{\eta}{1+\eta} \le 1/2$ for some $\eta \le 1$. For $i \in A$ we keep the same demand systems D_i and the same initial endowments $e^{(i)}$. The market $\hat{M}$ has an extra agent $n+1$ with initial endowment $e^{(n+1)} = \eta e$ (recall $e = \sum_{i\in A} e^{(i)}$) and whose demand bundle is given via the Cobb-Douglas utility function $\left(\prod_j x_j^{(n+1)}\right)^{1/m}$. Agent $n+1$ spends exactly $\frac{1}{m}$ of its budget on each good j since its unique demand bundle $x^{(n+1)}$ is given by $x_j^{(n+1)} = \frac{\eta\langle p,e\rangle}{mp_j}$.

The lemma below shows that adding such an agent can be used to bound $\frac{p_{\max}}{p_{\min}}$, at the expense of working on a modified market.

Lemma 12. *(i) For an ε-equilibrium of $\hat{M}$ formed by prices p and bundles $x^{(i)}$,*

$$\frac{p_{\max}}{p_{\min}} \le \frac{(1+\varepsilon)m}{\eta - \varepsilon(1+\varepsilon)(1+\eta)m}\cdot\frac{e_{\max}}{e_{\min}},$$

where $e_{\max} = \max_j e_j$ and $e_{\min} = \min_j e_j$.

(ii) An ε-equilibrium in $\hat{M}$ gives an $\varepsilon(1+\eta)$-equilibrium in M.

Proof. Consider an ε-equilibrium in $\hat{M}$ formed by prices p and bundles $x^{(i)}$, $i = 1, 2, \ldots, n+1$. By definition, there exist prices $p \le p^{(n+1)} \le (1+\varepsilon)p$ and bundle $z^{(n+1)} \in D_{n+1}\left(p^{(n+1)}, \eta\langle p, e\rangle\right)$ such that $x^{(n+1)} \le z^{(n+1)}$. We have $z_j^{(n+1)} = \frac{\eta\langle p,e\rangle}{mp_j^{(n+1)}}$, and therefore, $p_j z_j^{(n+1)} \ge \frac{\eta}{(1+\varepsilon)m}\langle p, e\rangle$. On the other hand, from the third condition of the definition of ε-equilibrium it follows that $p_j(z_j^{(n+1)} - x_j^{(n+1)}) \le \varepsilon(1+\eta)\langle p, e\rangle$. Hence, $p_j x_j^{(n+1)} \ge \left(\frac{\eta}{(1+\varepsilon)m} - \varepsilon(1+\eta)\right) \cdot \langle p, e\rangle$ for all j. In particular, $x_j^{(n+1)} \ge \left(\frac{\eta}{(1+\varepsilon)m} - \varepsilon(1+\eta)\right)\frac{p_{\max}e_{\min}}{p_j}$ for all j. Since $x_j^{(n+1)} \le e_j \le e_{\max}$ in an ε-equilibrium, we have

$$\frac{p_{\max}}{p_{\min}} \le \left(\frac{\eta}{(1+\varepsilon)m} - \varepsilon(1+\eta)\right)^{-1}\frac{e_{\max}}{e_{\min}}.$$

The second part of the lemma follows easily from the definition of an approximate equilibrium. □

ACKNOWLEDGMENTS.

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

REFERENCES

[1] Nima Anari, Tung Mai, Shayan Oveis Gharan, and Vijay V Vazirani. 2018. Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. In *Proceedings of the 29th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2274–2290.

[2] Gabriel Andrade, Rafael Frongillo, Sharadha Srinivasan, and Elliot Gorokhovsky. 2021. Graphical Economies with Resale. In *Proceedings of the 2021 ACM Conference on Economics and Computation (EC)*. 71–90.

[3] Kenneth J Arrow, Henry D Block, and Leonid Hurwicz. 1959. On the stability of the competitive equilibrium, II. *Econometrica: Journal of the Econometric Society* (1959), 82–109.

[4] Kenneth J Arrow and Gerard Debreu. 1954. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society* (1954), 265–290.

[5] Kenneth J Arrow and Leonid Hurwicz. 1958. On the stability of the competitive equilibrium, I. *Econometrica: Journal of the Econometric Society* (1958), 522–552.

[6] Kenneth J Arrow and Leonid Hurwicz. 1960. Competitive stability under weak gross substitutability: The "Euclidean distance" approach. *International Economic Review* 1, 1 (1960), 38–49.

[7] Noa Avigdor-Elgrabli, Yuval Rabani, and Gala Yadgar. 2014. Convergence of Tâtonnement in Fisher Markets. *arXiv preprint arXiv:1401.6637* (2014).

[8] Xiaohui Bei, Jugal Garg, and Martin Hoefer. 2019. Ascending-Price Algorithms for Unknown Markets. *ACM Transactions on Algorithms (TALG)* 15, 3 (2019), 37:1–37:33.

[9] Dimitri P Bertsekas. 1981. A new algorithm for the assignment problem. *Mathematical Programming* 21, 1 (1981), 152–171.

[10] Dimitri P Bertsekas. 1990. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces* 20, 4 (1990), 133–149.

[11] Benjamin Birnbaum, Nikhil Devanur, and Lin Xiao. 2011. Distributed algorithms via gradient descent for Fisher markets. In *Proceedings of the 12th Conference on Electronic Commerce (EC)*. 127–136.

[12] William C Brainard and Herbert E Scarf. 2005. How to compute equilibrium prices in 1891. *American Journal of Economics and Sociology* 64, 1 (2005), 57–83.

[13] Simina Brânzei, Nikhil R. Devanur, and Yuval Rabani. 2021. Proportional Dynamics in Exchange Economies. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. 180–201.

[14] Simina Brânzei, Ruta Mehta, and Noam Nisan. 2018. Universal Growth in Production Economies. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*. 1975.

[15] Bhaskar Ray Chaudhury, Yun Kuen Cheung, Jugal Garg, Naveen Garg, Martin Hoefer, and Kurt Mehlhorn. 2018. On Fair Division for Indivisible Items. In *Proceedings of the 38th IARCS annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. Springer, 25:1–25:17.

[16] Xi Chen, Decheng Dai, Ye Du, and Shang-Hua Teng. 2009. Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In *Proceedings of the 50th Symposium Foundations of Computer Science (FOCS)*. IEEE, 273–282.

[17] Yun Kuen Cheung, Richard Cole, and Nikhil R Devanur. 2019. Tâtonnement beyond gross substitutes? Gradient descent to the rescue. *Games and Economic Behavior* 123 (2019), 295–326.

[18] Yun Kuen Cheung, Richard Cole, and Ashish Rastogi. 2012. Tatonnement in ongoing markets of complementary goods. In *Proceedings of the 2012 ACM Conference on Electronic Commerce (EC)*. 337–354.

[19] Yun Kuen Cheung, Richard Cole, and Yixin Tao. 2018. Dynamics of Distributed Updating in Fisher Markets. In *Proceedings of the 19th ACM Conference on Economics and Computation*. 351–368.

[20] Yun Kuen Cheung, Martin Hoefer, and Paresh Nakhe. 2019. Tracing Equilibrium in Dynamic Markets via Distributed Adaptation. In *Proceedings of the 18th Conf. Auton. Agents and Multi-Agent Systems (AAMAS)*. 1225–1233.

[21] Bruno Codenotti, Benton McCune, and Kasturi Varadarajan. 2005. Market equilibrium via the excess demand function. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC)*. ACM, 74–83.

[22] Bruno Codenotti, Sriram Pemmaraju, and Kasturi Varadarajan. 2004. The computation of market equilibria. *ACM SIGACT News* 35, 4 (2004), 23–37.

[23] Bruno Codenotti, Sriram Pemmaraju, and Kasturi Varadarajan. 2005. On the polynomial time computation of equilibria for certain exchange economies. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 72–81.

[24] Bruno Codenotti, Amin Saberi, Kasturi R. Varadarajan, and Yinyu Ye. 2006. Leontief economies encode nonzero sum two-player games. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithmss*. 659–667.

[25] Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V Vazirani, and Sadra Yazdanbod. 2017. Convex Program Duality, Fisher Markets, and Nash Social Welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation (EC)*. ACM, 459–460.

[26] Richard Cole and Lisa Fleischer. 2008. Fast-converging tatonnement algorithms for one-time and ongoing market problems. In *Proceedings of the 40th ACM Symposium on Theory of Computing (STOC)*. ACM, 315–324.

[27] Richard Cole and Vasilis Gkatzelis. 2018. Approximating the Nash Social Welfare with Indivisible Items. *SIAM J. Comput.* 47, 3 (2018), 1211–1236.

[28] Gabrielle Demange, David Gale, and Marilda Sotomayor. 1986. Multi-item auctions. *Journal of Political Economy* 94, 4 (1986), 863–872.

[29] Nikhil Devanur, Christos Papadimitriou, Amin Saberi, and Vijay Vazirani. 2008. Market equilibrium via a primal–dual algorithm for a convex program. *Journal of the ACM* 55, 5 (2008).

[30] Nikhil R Devanur and Vijay V Vazirani. 2003. An improved approximation scheme for computing Arrow-Debreu prices for the linear case. In *Proceedings of the 23rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. Springer, 149–155.

[31] Nikhil R Devanur and Vijay V Vazirani. 2004. The spending constraint model for market equilibrium: Algorithmic, existence and uniqueness results. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, Vol. 36. ACM, 519–528.

[32] Ran Duan and Kurt Mehlhorn. 2015. A Combinatorial Polynomial Algorithm for the Linear Arrow-Debreu Market. *Information and Computation* 243 (2015), 112–132.

[33] Edmund Eisenberg. 1961. Aggregation of utility functions. *Management Science* 7, 4 (1961), 337–350.

[34] Edmund Eisenberg and David Gale. 1959. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics* 30, 1 (1959), 165–168.

[35] Lisa Fleischer, Rahul Garg, Sanjiv Kapoor, Rohit Khandekar, and Amin Saberi. 2008. A fast and simple algorithm for computing market equilibria. In *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*. Springer, 19–30.

[36] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. 2018. Approximating the Nash social welfare with budget-additive valuations. In *Proceedings of the 29th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2326–2340.

[37] Jugal Garg, Edin Husić, and László Végh. 2021. Auction Algorithms for Market Equilibrium with Weak Gross Substitute Demands and their Applications. In *Proceedings of the 38th Symposium on Theoretical Aspects of Computer Science (STACS)*.

[38] Jugal Garg, Edin Husić, and László A Végh. 2019. Auction algorithms for market equilibrium with weak gross substitute demands and their applications. *arXiv preprint arXiv:1908.07948* (2019).

[39] Jugal Garg, Ruta Mehta, Vijay V Vazirani, and Sadra Yazdanbod. 2017. Settling the complexity of Leontief and PLC exchange markets under exact and approximate equilibria. In *Proceedings of the 49th ACM Symposium on Theory of Computing (STOC)*. ACM, 890–901.

[40] Jugal Garg and László A Végh. 2022. A Strongly Polynomial Algorithm for Linear Exchange Markets. *Operations Research* (2022).

[41] Rahul Garg and Sanjiv Kapoor. 2006. Auction Algorithms for Market Equilibrium. *Mathematics of Operations Research* 31, 4 (2006), 714–729.

[42] Rahul Garg and Sanjiv Kapoor. 2006. Price roll-backs and path auctions: An approximation scheme for computing the market equilibrium. In *Proceedings of the 2nd International Workshop on Internet and Network Economics (WINE)*. Springer, 225–238.
[43] Rahul Garg and Sanjiv Kapoor. 2007. Market equilibrium using auctions for a class of gross-substitute utilities. In *Proceedings of the 3rd International Workshop on Web and Internet Economics (WINE)*. Springer, 356–361.
[44] Rahul Garg, Sanjiv Kapoor, and Vijay Vazirani. 2004. An auction-based market equilibrium algorithm for the separable gross substitutability case. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 128–138.
[45] Mehdi Ghiyasvand and James B Orlin. 2012. A simple approximation algorithm for computing Arrow-Debreu prices. *Operations Research* 60, 5 (2012), 1245–1248.
[46] Faruk Gul and Ennio Stacchetti. 1999. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory* 87, 1 (1999), 95–124.
[47] Faruk Gul and Ennio Stacchetti. 2000. The English auction with differentiated commodities. *Journal of Economic Theory* 92, 1 (2000), 66–95.
[48] K. Jain and K. Varadarajan. 2006. Equilibria for economies with production: Constant-returns technologies and production planning constraints. In *Proceedings of the 17th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 688–697.
[49] Sanjiv Kapoor, Aranyak Mehta, and Vijay Vazirani. 2007. An auction-based market equilibrium algorithm for a production model. *Theoretical Computer Science* 378, 2 (2007), 153–164.
[50] Wouter J. Keller. 1976. A nested CES-type utility function and its demand and price-index functions. *European Economic Review* 7 (1976), 175–186.
[51] Alexander S Kelso Jr and Vincent P Crawford. 1982. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society* (1982), 1483–1504.
[52] Renato Paes Leme. 2017. Gross substitutability: An algorithmic survey. *Games and Economic Behavior* 106 (2017), 294–316.
[53] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. 1995. *Microeconomic theory*. Vol. 1. Oxford university press New York.
[54] Kiminori Matsuyama and Philip Ushchev. 2017. Beyond CES: three alternative classes of flexible homothetic demand systems. *Global Poverty Research Lab Working Paper* 17-109 (2017).
[55] Lionel McKenzie. 1954. On equilibrium in Graham's model of world trade and other competitive systems. *Econometrica: Journal of the Econometric Society* (1954), 147–161.
[56] Lionel W McKenzie. 1959. On the existence of general equilibrium for a competitive market. *Econometrica: Journal of the Econometric Society* (1959), 54–71.
[57] Nimrod Megiddo and Christos H Papadimitriou. 1991. On total functions, existence theorems and computational complexity. *Theoretical Computer Science* 81, 2 (1991), 317–324.
[58] Yurii Nesterov and Vladimir Shikhman. 2018. Computation of Fisher–Gale Equilibrium by Auction. *Journal of the Operations Research Society of China* 6, 3 (2018), 349–389.
[59] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. 2007. *Algorithmic game theory*. Cambridge University Press.
[60] Paul A. Samuelson. 1947. *Foundations of Economic Analysis*. Harvard University Press.
[61] Herbert Scarf. 1960. Some examples of global instability of the competitive equilibrium. *International Economic Review* 1, 3 (1960), 157–172.
[62] Vijay Vazirani and Mihalis Yannakakis. 2011. Market equilibrium under separable, piecewise-linear, concave utilities. *Jounal of the ACM* 58, 3 (2011), 10.
[63] Léon Walras. 1896. *Éléments d'économie politique pure, ou, Théorie de la richesse sociale*. F. Rouge.
[64] Fang Wu and Li Zhang. 2007. Proportional response dynamics leads to market equilibrium. In *Proceedings of the 39th Symposiumn on Theory of Computing (STOC)*. 354–363.
[65] Yinyu Ye. 2008. A path to the Arrow-Debreu competitive market equilibrium. *Mathematical Programming* 111, 1-2 (2008), 315–348.
[66] Li Zhang. 2011. Proportional response dynamics in the Fisher market. *Theoretical Computer Science* 412, 24 (2011), 2691–2698.

Received April 2023; accepted August 2023