Ascending-Price Algorithms for Unknown Markets

XIAOHUI BEI, Nanyang Technological University
JUGAL GARG, University of Illinois at Urbana-Champaign
MARTIN HOEFER, Goethe University Frankfurt

We design a simple ascending-price algorithm to compute a $(1+\varepsilon)$-approximate equilibrium in Arrow-Debreu markets with weak gross substitute property. It applies to an unknown market setting without exact knowledge about the number of agents, their individual utilities, and endowments. Instead, our algorithm only uses price queries to a global demand oracle. This is the first polynomial-time algorithm for most of the known tractable classes of Arrow-Debreu markets, which computes such an equilibrium with a number of calls to the demand oracle that is polynomial in $\log 1/\varepsilon$ and avoids heavy machinery such as the ellipsoid method.

Demands can be real-valued functions of prices, but the oracles only return demand values of bounded precision. Due to this more realistic assumption, precision and representation of prices and demands become a major technical challenge, and we develop new tools and insights that may be of independent interest. Furthermore, we give the first polynomial-time algorithm to compute an exact equilibrium for markets with spending constraint utilities. This resolves an open problem posed by Duan and Mehlhorn.

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

Additional Key Words and Phrases: Market equilibrium, equilibrium computation, weak gross substitutes, spending constraint utilities

ACM Reference format:
Xiaohui Bei, Jugal Garg, and Martin Hoefer. 2019. Ascending-Price Algorithms for Unknown Markets. *ACM Trans. Algorithms* 15, 3, Article 37 (May 2019), 33 pages.
https://doi.org/10.1145/3319394

1 INTRODUCTION

The concept of market equilibrium is central in economics and captures fair, stable, and efficient outcomes in competitive allocation scenarios. The most prominent model to study market equilibria are *exchange markets* [2], which consist of a set of divisible goods and a set of agents. Each agent has an initial endowment of goods and a utility (preference) function over bundles of goods.

A one-page abstract of this work appeared in the proceedings of ACM EC'16 [4].
Jugal Garg was supported by NSF CRII Award 1755619.
Authors' addresses: M. Hoefer, Institute for Computer Science, Goethe University Frankfurt, Robert-Mayer-Strasse 11-15, 60325 Frankfurt am Main, Germany; email: mhoefer@cs.uni-frankfurt.de; J. Garg, Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, 104 S Mathews Ave, Urbana, IL 61801, USA; email: jugal@illinois.edu; X. Bei, Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, 637371; email: xhbei@ntu.edu.sg.

1549-6325/2019/05-ART37 $15.00
https://doi.org/10.1145/3319394

Given prices of goods, each agent buys a most preferred bundle (called *demand*) that is affordable from the earned money. At equilibrium, market clears (i.e., demand meets supply).

The computation of market equilibrium is a fundamental problem in economics and computer science [3, 27, 28, 45]. The challenge is to provide algorithms to compute an (approximate) market equilibrium efficiently. After more than a decade of research on this issue in theoretical computer science [9, 10, 16, 17, 22, 35, 50], there is a fairly good understanding of tractable and intractable domains (assuming P ≠ PPAD). For exchange markets, the tractable case is essentially[1] given by markets with a *weak gross substitute* (WGS) property, where any increase in prices of a set of goods does not strictly decrease the demand of unaffected goods. This includes markets with many popular and interesting classes of utility functions [41], such as utilities with constant elasticity of substitution (CES) with $0 < \rho < 1$, Cobb-Douglas utilities, linear utilities, or utilities with spending constraints[2] [24, 49].

Even beyond the economic interpretation, algorithms for computing market equilibria have wide applicability. For example, proportionally fair allocations, which result from market equilibria, are widely used in the design of computer networks [40]. Recent applications also include energy-efficient scheduling [38] and fair allocation of indivisible resources [1, 5, 18, 20, 32]. In the latter application, combinatorial polynomial-time algorithms are developed for variants of linear markets, for which no convex programming formulation is known.

A prerequisite for all of these efficient algorithms is the entire description of the market including number of agents, their utility functions, and initial endowments. Obtaining this description is a highly non-trivial task; an entire theory of *revealed preferences* [44, 47, 48] was developed to study how to infer market parameters from observed prices and buying patterns. Further, sometimes agents' preferences can be complicated, and their utility function may not have any succinct representation. This gives rise to the following question: Can we design efficient algorithms that are oblivious to *a priori* knowledge of market parameters? In other words, are there efficient algorithms for *unknown markets*, where one can learn the correct prices only via aggregate demand feedback?

In this work, we design a simple ascending-price algorithm for WGS exchange markets. Under the standard assumptions that the most preferred bundles are unique and continuous in all prices, our algorithm computes a strongly $(1+\varepsilon)$-approximate market equilibrium [17] in polynomial time (i.e., in time polynomial in market parameters and $\log 1/\varepsilon$). In these states, all buyers spend exactly their endowment. Moreover, for every buyer individually, it is guaranteed that she receives a bundle of goods with a utility that is at most a factor $(1+\varepsilon)$ worse than the optimal bundle. In addition, our algorithms only rely on aggregate demand feedback, and only require *polynomially bounded precision* in the numbers.

In markets with linear and spending constraint utilities, the uniqueness and continuity assumptions do not hold. Thus, even with tie-breaking rules that imply the WGS property, the existing algorithms for general WGS markets are not applicable. Instead, our algorithm can be adapted when using an appropriate tie-breaking rule for demands. In contrast to the FPTAS [24] for spending constraint exchange markets with a running time that depends on $1/\varepsilon^2$, the running time of our algorithm only grows with a factor of $\log 1/\varepsilon$. Notably, this $\log 1/\varepsilon$-dependence allows us to convert an approximate equilibrium to an exact one in polynomial time when given all utility parameters. Thus, we obtain the first polynomial-time algorithm for computing an *exact* market equilibrium for spending constraint utilities and settle an open question raised by Duan and Mehlhorn [26].

[1]The other tractable cases are CES utilities with $-1 \leq \rho < 0$ [14].

[2]Spending constraint utilities are a piecewise linear concave generalization of linear utilities, which satisfy many desirable properties and have many additional applications.

All of our results are achieved by a unifying framework that works directly on the price vector. It uses simple binary search to identify suitable multiplicative price updates for subsets of goods. As such, our approach is extremely easy to implement and, in particular, avoids black-box use of the ellipsoid method. Furthermore, by working on the full generality of exchange markets, our algorithms can be applied to the simpler case of *Fisher markets* [8].[3]

To compute approximate equilibria, our algorithms do not require access to an explicit description of the utility functions and endowments of the agents. Instead, the number of agents, the agents' endowments, and utilities all remain unknown. As in the case of tâtonnement algorithms [6, 7, 11, 12, 13, 19, 46], we assume that these parameters can only be queried implicitly via aggregate demand queries. In such a query, we present a non-negative price for each good. For each agent, the oracle translates the endowment of each agent into money and determines a utility-maximizing bundle of goods for the money. It then aggregates the demands for each good and returns the vector of aggregate demands as an answer to the query. Note that for many WGS markets, these demand oracles can be easily implemented. For example, for CES utilities, there are even closed-form formulas. For suitable tie breaking, we can even apply the algorithms to unknown markets with linear and spending constraint utilities, which have non-continuous demand. Perhaps surprisingly, our tâtonnement-style algorithms succeed to implicitly detect all relevant information and changes in preferences in an unknown market setting. This requires overcoming a number of technical challenges and introduction of new techniques that we discuss in more detail in the following.

Our approach overcomes significant restrictions in some previous tâtonnement-style algorithms to compute equilibria in exchange markets [16, 31, 36]. These algorithms either use the ellipsoid method and convex feasibility programs but need to add auxiliary buyers to the markets [16], or apply only to linear markets and rely on a careful assignment and reallocation for each individual buyer upon price change [36], or provide weak convergence guarantees in the *sum of all excess demands* over all goods but yield no guarantee on the (individual) optimality of the assigned bundles [31].

1.1 Model and Notation

Exact and approximate market equilibrium. In an exchange market, there is a set A of n *agents* and a set G of m *goods*. Each agent i has an initial endowment $w_{ij} \in \mathbb{R}_{\geq 0}$ of good j. By default, we use index i to denote agents and index j to denote goods throughout the article. We denote the total endowment of good j by $w_j = \sum_{i \in A} w_{ij}$. Without loss of generality, we choose the units of measurement such that $w_j = 1, \forall j \in G$. An *allocation* $\mathbf{x} = (x_{ij})_{i \in A, j \in G}$ is an assignment of goods to agents such that $x_{ij} \geq 0, \forall i \in A, j \in G$, and $\sum_i x_{ij} = 1$. Each agent i has a *utility function* $u_i(\mathbf{x}_i)$ that specifies the value agent receives from his bundle $\mathbf{x}_i$. An (exact) *market equilibrium* is a pair $(\mathbf{x}, \mathbf{p})$, where $\mathbf{x}$ is an allocation and $\mathbf{p} = (p_j)_{j \in G}$ is a vector of non-negative *prices* $p_j \geq 0$. In a market equilibrium, each agent obtains a budget of money by selling his endowment at the given prices. Then $\mathbf{x}_i$ represents a utility-maximizing bundle of goods that he can afford to buy at the current prices for his budget. We call such a bundle a *demand bundle* of agent i. In addition, we say $x_j = \sum_i x_{ij}$ is the *demand for good* j in allocation $\mathbf{x}$.

Definition 1.1. A bundle $\mathbf{x}_i$ is a *demand* of agent i at prices $\mathbf{p}$ if $u_i(\mathbf{x}_i) = \max\{u_i(\mathbf{y}_i) \mid \mathbf{y}_i \in \mathbb{R}^m_{\geq 0}, \mathbf{p}^T \mathbf{y}_i \leq \mathbf{p}^T \mathbf{w}_i\}$. A pair $(\mathbf{x}, \mathbf{p})$ is a *market equilibrium* if (1) $\mathbf{x}_i$ is a demand of agent i at $\mathbf{p}$, and (2) $\sum_i x_{ij} = 1$ for each good $j \in G$. If a pair $(\mathbf{x}, \mathbf{p})$ is not a market equilibrium, the *excess demand* $z_j = x_j - 1$ is non-zero for at least one good $j \in G$.

[3] Fisher markets are a subclass of exchange markets where buyers and sellers are different agents. Buyers bring money to buy goods, whereas sellers bring goods to earn money.

Let us consider a concept of *strong* approximate market equilibrium [16]. It relaxes only the market clearing constraint but not the condition that $\mathbf{x}_i$ is a demand for each agent i.

Definition 1.2. A pair $(\mathbf{p}, \mathbf{x})$ is a *μ-approximate equilibrium* ($\mu \geq 1$) if (1) for each agent i, $\mathbf{x}_i$ is a demand of agent i at prices $\mathbf{p}$, and (2) $\sum_i x_{ij} \leq \mu \sum_i w_{ij}$ for each good j.

In Fisher markets [8], agents are divided into buyers and sellers. Each buyer i comes with an initial endowment of money B_i. Each seller i has an initial endowment of goods $\mathbf{w}_i$. Each buyer has no value for money and is only interested in buying goods, and each seller is only interested in obtaining money. Fisher markets are a special case of exchange markets when we interpret money as a separate good.

Note that if we scale all prices in $\mathbf{p}$ by a positive constant $c > 0$, the demands stay unaffected, as income and spending scale exactly by c. Hence, the equilibrium conditions for exact and approximate equilibria are invariant under such scaling. In the classes of markets we study here, there is always an exact equilibrium with strictly positive prices. As such, we will normalize our price vectors throughout and assume that the smallest price $\min_j p_j = 1$.

Utility functions. Algorithms for computing market equilibria rely on structural properties of the utility functions. A natural class are *linear utilities* when each agent i has non-negative values $u_{ij} \geq 0$ for each good $j \in G$, and $u_i(\mathbf{x}) = \sum_{j \in G} u_{ij} x_{ij}$. As a generalization, we consider spending constraint utilities [24], where the utility derived by agent i from good j is given by a piecewise linear concave (PLC) function f_{ij}. The overall utility of agent i is additively separable among goods (i.e., $u_i(\mathbf{x}) = \sum_{j \in G} f_{ij}(x_{ij})$). Each f_{ij} is a PLC function with a number of linear segments. Each segment k has two parameters: the rate of utility u_{ijk} per unit of good derived on segment k and the maximum fraction[4] B_{ijk} of budget that can be spent on segment k.[5] All B_{ijk} are strictly positive, and concavity implies $u_{ijk} > u_{ij(k+1)}$. Here we assume that all u_{ijk}'s are integers, all B_{ijk}, w_{ij}'s are rational numbers, and the whole input can be represented in no more than L bits. Markets with spending constraint utilities have an equilibrium composed of rational numbers under mild sufficiency conditions (see Section 3 for details).

WGS markets. More generally, we consider non-decreasing utility functions that generate markets with the WGS property—when we increase a price of some good j, the demand for unaffected goods does not strictly decrease. If the demand function of each agent is continues and differentiable, then WGS property can be written as $\frac{\partial x_{j'}}{\partial p_j} \geq 0, \forall j \neq j'$. Note that this also implies that increasing the price of a subset of goods will result in the complementary set having non-decreasing demand, because one can increase the prices one by one and apply the WGS property sequentially. For general WGS markets, we will assume that all demand bundles of agents are unique. Prominent examples are markets with *Cobb-Douglas utilities* $u_i(\mathbf{x}) = \prod x_{ij}^{u_{ij}}$, or *CES utilities* $u_i(\mathbf{x}) = (\sum_{j \in G} u_{ij} x_{ij}^{\rho})^{1/\rho}$ with $u_{ij} > 0$ and $0 < \rho < 1$. Note that in such markets, even if all utility and endowment parameters are rationals of finite size, the demand bundles and the market equilibrium might involve irrational numbers. In this case, we are interested in approximate

[4]Since, unlike Fisher markets, the budget of an agent in exchange markets depends on the prices of goods, the second parameter of a segment is defined as the fraction of the agent's budget that can be spent on that segment. In addition, we note that the spending constraint utilities are different from the other well-studied separable piecewise linear concave (SPLC) utilities [33]. In SPLC, each linear segment k has two parameters: the rate of utility and the maximum *amount of good* that can be bought on k. However, unlike the spending constraint utilities, equilibrium computation in markets with SPLC utilities is PPAD-complete [9, 50].

[5]Suppose agent i decides to spend a fraction B_{ij} of his budget on good j. The budget constraint implies that the agent gets a utility u_{ij1} per unit of good j only for the amount bought by the first B_{ij1}-fraction of his money, then u_{ij2} per unit for amount bought by the next B_{ij2}-fraction of his money, and so on, until B_{ij} is reached.

market equilibria, and our prices will use a pre-specified precision depending on the desired approximation factor.

Oracles. Our algorithm queries demands for the agents by publishing prices $\mathbf{p}$. Then an oracle returns the total demand x_j for each good $j \in G$. It assumes that each agent can sell his initial endowment at the given prices and then requests a utility-maximizing bundle of goods for the money he has available. For ease of notation, given any price vector $\mathbf{p}$, let $O(\mathbf{p})$ denote the *surplus vector* $\mathbf{s} = (s_1, \ldots, s_m)$ for the return of the oracle, where $s_j = p_j z_j$ is the *surplus* (in terms of money) of good $j \in G$. In other words, assuming that we publish $\mathbf{p}$, the oracle returns an excess demand vector $\mathbf{z} = (z_1, \ldots, z_m)$, then[6] $O(\mathbf{p}) = \mathbf{p} \cdot \mathbf{z}$.

In general WGS markets, the surplus vector might contain irrational values. Thus, we use an *approximate demand oracle* $\widetilde{O}(\mathbf{p}, \mu)$, which is a blackbox algorithm that takes any price vector $\mathbf{p}$ and positive rational μ as input. It returns a surplus vector $\mathbf{s}$ such that $|s_i - O(\mathbf{p})_i| \le \mu$ holds for every good i. Note that for many WGS markets, the demand oracle can be implemented very quickly—for CES utilities, there are even closed-form expressions for the demand of each agent for each good as a function of prices, utility, and endowment. More generally, we assume that the oracle can be implemented in time polynomial in the input size and $\log 1/\mu$. This standard assumption for demand oracles has also been used in previous work [15, 16].

In linear and spending constraint markets, a major challenge for an algorithm in unknown markets is non-uniqueness of demands. Here, an oracle needs to do tie breaking between several different bundles of goods that yield the same maximum utility for an agent. Ideally, it should satisfy the following properties: (1) the output demand is always deterministic and unique; (2) if $\mathbf{p}$ are equilibrium prices, the output demand equals supply for every good; and (3) the oracle can be implemented in time polynomial in the input size. Based on these criteria, we use a demand oracle that yields demands minimizing the ℓ_2-norm of the surplus vector. More formally, for prices $\mathbf{p}$, our oracle returns a set of demand bundles for the agents such that $\sum_i s_i^2$ is minimized, where s_i is the surplus of good i. Such a tie-breaking rule was introduced by Devanur et al. [23], and it has been later used in several market algorithms, such as those presented in other works [25, 26, 49]. This oracle satisfies all three properties. Furthermore, if utilities, endowments, and prices are all given as integers with a number of bits polynomial in m and L, we can represent every surplus s_i also with a number of bits polynomial in m and L. Hence, by setting μ sufficiently small, we can convert an approximate demand oracle $\widetilde{O}$ into an exact oracle in polynomial time. Therefore, we will assume that in spending constraint markets, we are equipped with an exact demand oracle O instead of an approximate one.

1.2 Results and Contribution

We present simple ascending-price algorithms that converge to market equilibrium in WGS and spending constraint markets. Our algorithm for WGS markets in Section 2 converges to a $(1+\varepsilon)$-approximate equilibrium in time polynomial in m, $\log 1/\varepsilon$, and other market parameters. In our algorithm, the number of agents, the agents' endowment, and utilities are initially all unknown. We only query a global demand oracle that provides aggregate demands for goods at given prices. This information is then used to increase prices of a selected set of goods whose demand is more than their supply. In Section 2, we ignore for simplicity all precision and representation issues to highlight the general proof technique. The complete analysis with all details of our algorithm is provided in Appendix A, where we specify in advance a precision for prices and rely on

[6]We define the surplus vector as price times excess demand because it satisfies $\sum_i s_i = 0$. This invariant is useful in design and analysis of our algorithms in the following.

approximate demand oracles whose output is within our desired bit precision. One can view our algorithm as a particular form of tâtonnement. It improves over previous approaches for WGS markets [15, 31] by decreasing the dependence of the running time to $\log 1/\varepsilon$, by working directly with aggregate demands without transformations like adding auxiliary agents, and by providing a strong guarantee on the optimality of individual bundles.

Next, we apply our approach in Section 3 to spending constraint utilities—a PLC generalization of linear utilities, which has many additional applications due to its natural diminishing returns property. Since these markets have demands that are non-continuous in the prices, we cannot directly apply our algorithm or other previous algorithms for WGS markets. Instead, we adjust our approach to implicitly capture the non-continuity events when using a global demand oracle with suitable tie breaking. When all parameters are represented by at most L bits, our algorithm computes even an exact market equilibrium in time polynomial in m and L. All prices and demands occurring during the algorithm require a bit precision polynomial in m and L. It first computes a $(1+\varepsilon)$-approximate equilibrium using a precision that is polynomial in m, L, and $\log 1/\varepsilon$. The exact demands returned by the demand oracle have the same precision. For a small enough ε (using only polynomial bit length), we can then use a rounding procedure to turn it into a price vector of an exact market equilibrium.

Note that our algorithm requires only access to a suitable demand oracle to compute an approximate equilibrium. However, in contrast to WGS markets, for spending constraint markets the oracle uses global tie breaking. To obtain an exact equilibrium, our final rounding procedure relies on full information about the utilities. Thus, we obtain in polynomial time an approximate equilibrium in unknown markets (with global tie breaking) and an exact equilibrium with full information. This represents the first polynomial-time algorithm to compute an exact market equilibrium for spending constraint utilities and settles an open question raised by Duan and Mehlhorn [26]. An important open problem is to construct efficient tâtonnement algorithms that avoid global tie breaking and full-information rounding.

We use and extend ideas of algorithms for linear markets with full information [23, 26]. These ideas were also used in spending constraint markets [24] to compute a $(1+\varepsilon)$-approximate equilibrium in time polynomial in m, L, and $1/\varepsilon$. Roughly speaking, they are intimately tied to linear and spending constraint utilities, where they work on the agents' side and increase prices until structural changes occur in the optimal bang-per-buck relations. The progress toward equilibrium is measured in the reduction of the ℓ_2-norm of surplus. Our approach works for all WGS markets, which generally do not exhibit such structural events that can be used for analysis. This turns out to be much more demanding, and we developed an approach that works entirely on the goods' side, based only on prices and aggregate demands obtained via oracle access.

An issue that plays a central role in our algorithms for spending constraint utilities is precision of prices and demands. This seems to have been treated only in minor detail in some of the previous works. For the algorithm of Duan and Mehlhorn [26] in linear exchange markets, these issues are discussed in depth. Their solution is to change the agents' side and alter utility values for maintaining bounded precision throughout the algorithm. However, computing an approximate equilibrium in unknown markets with demand oracle access seems impossible via this route.

As a consequence, unlike for the existing algorithms in the linear case, the surpluses encountered by our algorithms might now become negative. Hence, additional events have to be taken into account upon increasing prices. Moreover, a significant challenge lies in maintaining the precision of prices to be polynomial throughout the algorithm. To overcome this problem, we make use of a novel tool that we term the *ratio graph*. This graph is defined for a vector of prices $\mathbf{p}$. The goods are the vertices, and we draw an undirected edge between goods j and k if the ratio of prices p_j/p_k can be expressed by two L-bit numbers. For an intuition, observe that if some agent i has the same bang per buck for two goods j and k, then $u_{ij}/p_j = u_{ik}/p_k$ or $p_j/p_k = u_{ij}/u_{ik}$ (i.e., the ratio of prices can be expressed by two L-bit numbers). Maybe surprisingly, the broad structure of the ratio graph indeed contains enough information to implement algorithms for finding approximate equilibria in unknown spending constraint markets.

1.3 Related Work

The problem of computing market equilibria has been intensively studied, and the literature is too vast to survey here. We provide an overview of the work most directly relevant to ours. There is a large body of work on algorithms for computing equilibrium using full market information. The first combinatorial polynomial-time algorithm for linear Fisher markets was given by Devanur et al. [23]. Later, Vazirani [49] provided a polynomial-time algorithm for Fisher markets with spending constraint utilities by extending combinatorial techniques of Devanur et al. [23]. Strongly polynomial-time algorithms are also known for Fisher markets with linear [43, 51] and spending constraint utilities [51]. A simplex-like polynomial time algorithm to compute an approximate equilibrium for spending constraint utilities is given in Garg et al. [34].

For linear exchange markets, Jain [39] and Ye [53] obtained polynomial-time algorithms based on ellipsoid and interior point methods on a convex program, respectively. Duan and Mehlhorn [26] gave the first combinatorial polynomial-time algorithm for this problem, which was recently improved by Duan et al. [25]. For exchange markets with spending constraint utilities, Devanur and Vazirani [24] gave an algorithm to compute a $(1 + \epsilon)$-approximate equilibrium, for which the running time dependence on ϵ is $O(1/\epsilon^2)$. Eaves [28] gave a strongly polynomial-time algorithm for markets with Cobb-Douglas utilities.

For the general case of WGS markets with unique and continuous demands, a polynomial-time algorithm was obtained by Codenotti et al. [16]. Note that this algorithm relies heavily on the ellipsoid method. For the Fisher setting, the famous Eisenberg-Gale convex program [30] captures market equilibrium under linear utilities. Eisenberg [29] generalized it to work for any homogeneous utility functions, many of which satisfy the WGS property.

As an alternative approach to compute market equilibrium, *tâtonnement* was defined by Walras [52]—algorithms that have access to endowment and utilities only via demand oracles. Usually, tâtonnement procedures conduct price updates separately for each good, sometimes based on derivatives of the demand as a function of price. In the computer science literature, Codenotti et al. [15] gave a discrete tâtonnement process that converges to an $(1 + \epsilon)$-approximate equilibrium for WGS markets. It has a convergence time polynomial in the input size and $1/\epsilon$, and it does not query the original market because one needs to add auxiliary agents. By taking a different approach, our algorithm for the same market setting improves this rate to polynomial in the input size and $\log 1/\epsilon$. In addition, we only rely on approximate demand queries to the original market.

More recently, Cole and Fleischer [19] established the first fast-converging discrete version of tâtonnement for WGS markets. The convergence time depends on various market parameters. It requires a non-zero amount of money in the market, so it works for the special case of Fisher markets and beyond, but it is not applicable to the full range of exchange markets. For Fisher markets, many additional results [6, 11–13] on the convergence of tâtonnement processes beyond WGS markets were derived.

An auction-based ascending-price algorithm to compute an approximate equilibrium in linear exchange markets was obtained by Garg and Kapoor [36]. Its running time is polynomial in the input size and $1/\epsilon$. Later, it was generalized to a class of WGS markets for the case of Fisher markets [37]. Another tâtonnement-style algorithm is obtained by Fleischer et al. [31] for markets where the function of prices and income that gives the maximum utility achievable is convex. It was shown that this condition is satisfied by a certain class of Fisher markets that also include non-WGS markets. However, it provides weak convergence guarantees where budget constraints are satisfied only in the aggregate sense, in which some buyers may be spending significantly more than their budget.

2 WGS EXCHANGE MARKETS

In this section, we describe the algorithm for WGS exchange markets. As mentioned previously, we assume that we are only granted access to an approximate demand oracle and are restricted to finite precision arithmetic computations. To make our algorithm and its analysis more accessible, we will simplify the problem in the remainder of this section by assuming that we are equipped with (1) exact real arithmetic and (2) an exact demand oracle. This significantly simplifies the analysis in terms of notation and calculations, and as such concentrates on the key ideas of the algorithm. A complete and rigorous version of this section, presenting the entire algorithm and its proof with approximate precision, can be found in Appendix A.

We apply algorithm Alg-WGS-Precise given in the following. The main idea is to repeatedly identify a subset of goods G_1 by finding a gap in the sorted order of surpluses. If no such gap exists, let G_1 be the set of goods with positive surpluses. Note that in line 6 of the algorithm, the index k always exists because the surpluses of all goods add up to zero, which means the smallest surplus is always non-positive. Thus, set G_1 can always be identified. It then raises the prices of G_1 by a common factor x until the surplus gap is closed or the smallest surplus of G_1 becomes zero. More formally, given price vector $\mathbf{p} = (p_1, \ldots, p_m)$, value $x \in \mathbb{R}^+$ and subset $S \subseteq G$, the algorithm uses $\text{Update}(\mathbf{p}, x, S)$, which is the price vector $\mathbf{p}' = (p'_1, \ldots, p'_m)$ with $p'_i = x \cdot p_i$ if $i \in S$ and $p'_i = p_i$ otherwise. To implement this process, the algorithm relies on two parameters D_1 and D_2 based on the following assumptions.

Assumption 2.1 (Bounded Price). *There exists a market equilibrium* $(\mathbf{p}^*, \mathbf{x}^*)$ *with* $1 \leq p_i^* \leq 2^{D_1}, \forall i \in G$.

Assumption 2.2 (Continuity). *For any price vector* $\mathbf{p}$ *such that* $1 \leq p_i \leq 2^{D_1}$ *for each* i, $|\frac{\partial s_i}{\partial p_j}| < 2^{D_2}$ *for every* i, j*, where* s_i *is the surplus money of good* i *in* $\mathcal{O}(\mathbf{p})$*, and* D_2 *is a polynomial of the input size.*

These two assumptions are precisely the ones from Codenotti et al. [15, 16]. Assumption 2.1 about bounded prices is fairly mild and in many cases necessary for an efficient algorithm to compute a (strong) approximate market equilibrium. Assumption 2.2 about continuity is also satisfied by many natural markets, such as markets with CES utilities with $0 < \rho < 1$. Note, however, that it is not satisfied for linear and spending constraint markets, and hence we must develop new tools and procedures in Section 3.

To measure progress toward equilibrium, we use a *potential function* $\Phi(\mathbf{p}_t) = \|\mathcal{O}(\mathbf{p}_t)\|_2^2$. We start by proving a number of claims about the price vector $\mathbf{p}_t$. The first claim shows that with respect to exact demands, our algorithm monotonically reduces the 1-norm of the surpluses of all goods from $2m$.

Claim 2.1. *In Alg-WGS-Precise,* $|\mathcal{O}(\mathbf{p}_0)| \leq 2m$ *and* $|\mathcal{O}(\mathbf{p}_t)|$ *is non-increasing in* t.

Proof. Let d_i be the exact demand for good i under price $\mathbf{p}_0$, then $|\mathcal{O}(\mathbf{p}_0)| = \sum_i |d_i - 1| \leq \sum_i d_i + m = 2m$. Next, by the criteria to define G_1 and G_2 in each round, we have $\{i \mid \mathcal{O}(\mathbf{p}_{t-1})_i < 0\} \subseteq G_2$.

During round t, only prices of goods in G_1 are increased. By the WGS property, we know $\mathcal{O}(\mathbf{p}_t)_i \geq \mathcal{O}(\mathbf{p}_{t-1})_i$ for every $i \in G_2$. Further, note that $\min\{\mathcal{O}(\mathbf{p}_t)_i \mid i \in G_1\} \geq 0$ since $\min\{\mathcal{O}(\mathbf{p}_t)_i \mid i \in G_1\} \geq 0$. Hence, we do not introduce any new negative surplus in $\mathcal{O}(\mathbf{p}_t)$. Thus, we have

ALGORITHM 1: ALG-WGS-PRECISE

Input: number of goods m, demand oracle $\mathcal{O}$, precision bound $\varepsilon > 0$
Output: Prices $\mathbf{p}$ of an $(1+\varepsilon)$-approximate market equilibrium

1 Set initial price $\mathbf{p}_0 \leftarrow (1, 1, \ldots, 1)$ and round index $t \leftarrow 0$.
2 **repeat** //round t
3 $\quad t \leftarrow t + 1$
4 $\quad \mathbf{s} = (s_1, \ldots, s_m) \leftarrow \mathcal{O}(\mathbf{p}_{t-1})$
5 $\quad$ Sort $\mathbf{s}$ such that $s_{i_1} \geq s_{i_2} \geq \cdots \geq s_{i_m}$.
6 $\quad$ Find smallest k such that $s_{i_{k+1}} \leq 0$ or $s_{i_k} > (1 + \frac{1}{m}) s_{i_{k+1}}$.
7 $\quad$ Set $G_1 \leftarrow \{i_1, \ldots, i_k\}$ and $G_2 \leftarrow [m] \setminus G_1$
8 $\quad$ Find the largest x such that in $\mathbf{s}' = \mathcal{O}(\text{UPDATE}(\mathbf{p}_{t-1}, x, G_1))$, it holds
$\qquad \min\{s'_i \mid i \in G_1\} = \max\{\{s'_i \mid i \in G_2\} \cup \{0\}\}$.
9 $\quad \mathbf{p}_t \leftarrow \text{UPDATE}(\mathbf{p}_{t-1}, x, G_1)$
10 **until** $\|\mathcal{O}(\mathbf{p}_t)\|^2 < \varepsilon^2$
11 **return** $\mathbf{p}_t$

$$|\mathcal{O}(\mathbf{p}_{t-1})| = -2 \sum_{\mathcal{O}(\mathbf{p}_{t-1})_i < 0} \mathcal{O}(\mathbf{p}_{t-1})_i \geq -2 \sum_{\mathcal{O}(\mathbf{p}_t)_i < 0} \mathcal{O}(\mathbf{p}_{t-1})_i \geq -2 \sum_{\mathcal{O}(\mathbf{p}_t)_i < 0} \mathcal{O}(\mathbf{p}_t)_i = |\mathcal{O}(\mathbf{p}_t)|. \qquad \square$$

The next two claims bound the range of prices we encounter, which is important for showing that we approach the unique market equilibrium.

CLAIM 2.2. *Throughout the run of ALG-WGS-PRECISE, every good with negative surplus has price 1. Hence, there will be at least one good whose price remains 1.*

PROOF. Observe the following three simple facts about the surplus $\mathcal{O}(\mathbf{p}_t)$ resulting from exact demands. First, throughout the algorithm, we never increase the price of any good with negative surplus. Second, the surplus of any good does not change from non-negative to negative. Third, for any non-equilibrium price vector, there will always be a good with negative surplus. These facts are direct consequences of the conditions used to classify goods based on $\mathcal{O}(\mathbf{p}_t)$ in the algorithm. Together they prove the claim. $\square$

CLAIM 2.3. *In ALG-WGS-PRECISE, the x that satisfies the condition in Step 8 always exists. Further, for any $t \geq 0$, all prices in $\mathbf{p}_t$ are bounded by 2^{D_1}.*

PROOF. Let $\mathbf{p}^*$ be equilibrium prices according to Assumption 2.1. Let $t \geq 0$ be any round such that $\mathbf{p}_t$ is pointwise no larger than $\mathbf{p}^*$. Let G_1 be the set as defined in the algorithm in round t, and $x^* = \min_{i \in G_1} \{\frac{\mathbf{p}^*_i}{(\mathbf{p}_t)_i}\}$. In the following, we will show that with $\mathbf{p} = \text{UPDATE}(\mathbf{p}_t, x^*, G_1)$, there must exist some $i \in G_1$ such that $\mathcal{O}(\mathbf{p})_i \leq 0$. If this assertion is true, by the continuity Assumption 2.2 we know (1) there exists some $x \leq x^*$ that satisfies the condition in Step 8, and (2) $\mathbf{p}_{t+1} = \text{UPDATE}(\mathbf{p}_t, x, G_1)$ is pointwise no larger than $\mathbf{p}$, which in turn is pointwise no larger than $\mathbf{p}^*$. Thus, the claim holds by induction.

Assume that this assertion is not true—that is, $\mathcal{O}(\mathbf{p})_i > 0$ for every $i \in G_1$. Let $S = \{i \mid \mathbf{p}_i = \mathbf{p}^*_i\} \cap G_1$. Note that by the definition of x^*, S is non-empty, and $\mathbf{p}$ is still pointwise no larger than $\mathbf{p}^*$. Next, we increase $\mathbf{p}_j$ to $\mathbf{p}^*_j$ for every $j \notin S$. Then $\mathbf{p}$ becomes exactly $\mathbf{p}^*$. By the WGS property of the market, during this process the surplus of any good in S will not decrease, and hence we still have $\mathcal{O}(\mathbf{p}^*)_i \geq \mathcal{O}(\mathbf{p})_i > 0$ for every $i \in S$. This contradicts the assumption that $\mathbf{p}^*$ are prices of a market equilibrium. $\square$

The following claim establishes a relation between the surplus of a good with respect to exact demands before and after a multiplicative price update step.

Claim 2.4. *For any price vector* $\mathbf{p}$*,* $x > 1$ *and* $S \subseteq [m]$*,* $O(\textsc{Update}(\mathbf{p}, x, S))_i \leq x \cdot O(\mathbf{p})_i$ *for any* $i \in S$*.*

Proof. Because prices are scalable in exchange markets, we have $O(x \cdot \mathbf{p}) = x \cdot O(\mathbf{p})$ for any value $x > 0$. In addition, by the WGS property, when we decrease any set of prices, this will not increase the demand for unaffected goods. Since these goods have unaffected price and non-decreasing demand, they also enjoy non-decreasing surplus. Therefore, for any $i \in S$, we have $O(\textsc{Update}(\mathbf{p}, x, S))_i \leq O(x \cdot \mathbf{p})_i = x \cdot O(\mathbf{p})_i$. □

The next lemma is the key step in the proof of our main result. It establishes a multiplicative decrease of the potential function at the end of many of the rounds. Let $R = 48e^2$ in all following lemmas within this section.

Lemma 2.1. *If* $x < 1 + \frac{1}{Rm^3}$ *at the end of round* t *in* Alg-WGS-Precise*, then* $\Phi(\mathbf{p}_t) \leq (1 - \frac{1}{16e^2m^3})$ $\Phi(\mathbf{p}_{t-1})$*.*

Proof. We let $\mathbf{s} = O(\mathbf{p}_{t-1})$ and $\mathbf{s}' = O(\mathbf{p}_t)$ throughout the proof. An intuition of the proof is as follows. In Alg-WGS-Precise, by the conditions used to define G_1 and G_2, we always have $s_{i_1} \leq (1 + \frac{1}{m})s_{i_2} \leq \cdots \leq (1 + \frac{1}{m})^{k-1}s_{i_k} < e \cdot s_{i_k}$ and $s_{i_k} - s_{i_{k+1}} > s_{i_k}/(m+1) \geq s_{i_1}/e(m+1)$. Hence, roughly speaking, every good in G_1 has reasonably large surplus, and there is a reasonably large gap between the surpluses in G_1 and G_2. Next, at the end of the current round, we decreased the minimum surplus of a good in G_1 to either $\min\{s'_i \mid i \in G_1\} = 0$ (Case (1) shown later) or $\min\{s'_i \mid i \in G_1\} = \max\{s'_i \mid i \in G_2\}$ (Case (2) shown later). In both cases, the total value of Φ must decrease by at least a factor of $1 - 1/16e^2m^3$.

More formally, if the algorithm proceeds to round t, then $\|\mathbf{s}\| > \varepsilon^2$. By the definition of set G_1, we have $s_{i_1} \leq (1 + \frac{1}{m})s_{i_2} \leq \cdots \leq (1 + \frac{1}{m})^{k-1}s_{i_k} < e \cdot s_{i_k}$. Hence, $s_{i_k}^2 > (s_{i_1}/e)^2 \geq \Phi(\mathbf{p}_{t-1})/(me^2) > (\varepsilon/(\sqrt{m}e))^2$, so the surpluses of goods in G_1 are similar up to a factor of e and bounded from below. In addition, we have $(s_{i_k} - s_{i_{k+1}})^2 > (s_{i_1}/e(m+1))^2 \geq \Phi(\mathbf{p}_{t-1})/(e^2(m+1)^2m)$.

Next we relate the surplus in the beginning and the end of a round as follows. For every $i \in G_1$, by Claim 2.4, the surplus from exact demands satisfies $s'_i \leq x \cdot s_i$. Since $x < 1 + \frac{1}{Rm^3}$, it holds that $s'_i < (1 + \frac{1}{Rm^3})s_i$. We do not touch the price of any good $j \in G_2$, so the WGS property implies for exact demands $s'_j \geq s_j$.

Now, to bound the change of $\Phi(\mathbf{p}_t)$, we consider $\mathbf{s}'$ according to G_1 and G_2. We distinguish two cases:

Case 1: $\max\{s'_i \mid i \in G_2\} < 0$. In this case, the algorithm has decreased the surplus of some good in G_1 to 0. This decrease alone brings down the potential function Φ by a factor of $1 - \Omega(1/m)$. All other surpluses will cause an increase by a factor of at most $1 + O(1/m^3)$.

More formally, the contribution of goods of G_1 to $\Phi(\mathbf{p}_t)$ can be upper bounded by

$$\sum_{j=1}^{k} s'^2_{i_j} < \sum_{j=1}^{k-1} \left(1 + \frac{1}{Rm^3}\right)^2 s_{i_j}^2. \tag{1}$$

Furthermore, for every $i \in G_2$, by the WGS property of the market, we know $s_i \leq s'_i < 0$. Thus, the contribution of goods of G_2 to $\Phi(\mathbf{p}_t)$ can be upper bounded by

$$\sum_{j=k+1}^{m} s'^2_{i_j} \leq \sum_{j=k+1}^{m} s_{i_j}^2 < \sum_{j=k+1}^{m} \left(1 + \frac{1}{Rm^3}\right)^2 s_{i_j}^2. \tag{2}$$

Combining equations (1) and (2),

$$\begin{aligned}
\Phi(\mathbf{p}_t) = \sum_{j=1}^{m} s'^2_{i_j} &< \sum_{j\neq k}\left(1+\frac{1}{Rm^3}\right)^2 s^2_{i_j} \\
&= \left(1+\frac{1}{Rm^3}\right)^2\left(\Phi(\mathbf{p}_{t-1}) - s^2_{i_k}\right) \\
&< \left(1+\frac{1}{Rm^3}\right)^2\left(1-\frac{1}{e^2m}\right)\Phi(\mathbf{p}_{t-1}) \\
&< \left(1+\frac{3}{Rm^3}\right)\left(1-\frac{1}{e^2m}\right)\Phi(\mathbf{p}_{t-1}) \\
&< \left(1-\frac{1}{2e^2m}\right)\Phi(\mathbf{p}_{t-1}),
\end{aligned}$$

where the last two inequalities hold for any $m \geq 2$ with our choice of the value of R.

Case 2: $\max\{s'_i \mid i \in G_2\} \geq 0$. In this case, the gap between surpluses in G_1 and G_2 decreases to 0. In the following, we show that the closing of this gap yields a decrease of the potential function Φ by a factor of $1 - \Omega(1/m^3)$. All other surpluses will increase by a factor of at most $1 + O(1/m^3)$. In combination, it turns out that Φ will decrease by a factor of $1 - \Omega(1/m^3)$.

More formally, in this case, $\min\{s'_i \mid i \in G_1\} = \max\{s'_j \mid j \in G_2\}$. Let $s_{G_1} = \min\{s'_i \mid i \in G_1\}$ and $s_{G_2} = \max\{s'_j \mid j \in G_2\}$. For every $i \in G_1$, let $s'_i = x' s_i - \delta_i$, where $x' = (1 + \frac{1}{Rm^3})$, and for every $j \in G_2$, let $s'_j = s_j + \delta_j$. Hence, $\delta_i, \delta_j \geq 0$ for all i, j. Further, we have $\sum_{i=1}^m s_i = \sum_{i=1}^m s'_i = 0$, and hence

$$\begin{aligned}
\sum_{i\in G_1}\delta_i &= \sum_{i\in G_1} x's_i - \sum_{i\in G_1} s'_i \\
&\geq \sum_{i\in G_1} s_i + \sum_{j\in G_2} s'_j = \sum_{i\in G_1} s_i + \sum_{j\in G_2}(s_j+\delta_j) \\
&= \sum_{j\in G_2}\delta_j
\end{aligned}$$

and

$$\sum_{i\in G_1}\delta_i \geq \frac{1}{2}(s_{i_k} - s_{i_{k+1}}).$$

Now we have

$$\begin{aligned}
\Phi(\mathbf{p}_t) = \sum_i s'^2_i &= \sum_{i\in G_1}(x's_i-\delta_i)^2 + \sum_{j\in G_2}(s_j+\delta_j)^2 \\
&= \left(\sum_{i\in G_1} x'^2 s_i^2 + \sum_{j\in G_2} s_j^2\right) + \left(\sum_{j\in G_2}\delta_j(s_j+\delta_j) - \sum_{i\in G_1}\delta_i(x's_i-\delta_i)\right) - \sum_{i\in G_1} x's_i\delta_i + \sum_{j\in G_2}\delta_j s_j \quad (3) \\
&< x'^2\Phi(\mathbf{p}_{t-1}) + \left(s_{G_2}\sum_{j\in G_2}\delta_j - s_{G_1}\sum_{i\in G_1}\delta_i\right) - s_{i_k}\sum_{i\in G_1}\delta_i + s_{i_{k+1}}\sum_{j\in G_2}\delta_j \quad (4) \\
&< x'^2\Phi(\mathbf{p}_{t-1}) - (s_{i_k} - s_{i_{k+1}})\sum_{i\in G_1}\delta_i \quad (5) \\
&< x'^2\Phi(\mathbf{p}_{t-1}) - \frac{1}{2}(s_{i_k} - s_{i_{k+1}})^2 \quad (6)
\end{aligned}$$

$$< \left(1 + \frac{2}{Rm^3} + \frac{1}{R^2m^6} - \frac{1}{2e^2(m+1)^2m}\right)\Phi(\mathbf{p}_{t-1}) \quad (7)$$

$$< \left(1 + \frac{3}{Rm^3} - \frac{1}{8e^2m^3}\right)\Phi(\mathbf{p}_{t-1}) \quad (8)$$

$$= \left(1 - \frac{1}{16e^2m^3}\right)\Phi(\mathbf{p}_{t-1}). \quad (9)$$

Here, (3) can be derived by expanding the quadratic formula and appropriately reorganizing the terms. For the step from (3) to (4), in the first bracket we overestimate the quadratic terms of s into $x'^2\Phi(\mathbf{p}_{t-1})$. In the second bracket, the terms are bounded correctly using s_{G_1} for all $i \in G_1$ and s_{G_2} for all $j \in G_2$. For the final two terms in (3) and (4), we use the definition of s_{i_k} and $s_{i_{k+1}}$ and the fact that $x' > 1$. For the step from (4) to (5), for the second bracket of (4) we note $s_{G_1} \geq s_{G_2}$ and the two sets G_1 and G_2 have the same sums of δ-terms. By the same argument, we can transform the last two terms of (4) as shown. From (5) to (6) and then to (7), we use the bound for $\sum_{i \in G_1} \delta_i$ and $(s_{i_k} - s_{i_{k+1}})^2 > \Phi(\mathbf{p}_{t-1})/(e^2(m+1)^2m)$ as shown earlier.

Finally, the multiplicative term in (7) can be decreased to strictly less than 1 for every $m \geq 2$ by our choice of the value of R. The final expression in (9) proves the lemma. □

Observe that the previous lemma shows a decrease in the potential only for rounds in which the factor x is rather small. The next lemma shows that there can be only a limited number of rounds with a larger value of x.

Lemma 2.2. *During a run of Alg-WGS-Precise, there can be only $O(m^4D_1)$ many rounds that end with $x \geq 1 + \frac{1}{Rm^3}$.*

Proof. By Claim 2.3, every price can be increased by a factor of $1 + \frac{1}{Rm^3}$ at most $O(\log_{1+1/Rm^3} 2^{D_1}) = O(m^3D_1)$ times. Hence, there can be at most $O(m^4D_1)$ many rounds with $x \geq 1 + \frac{1}{Rm^3}$. □

Finally, we can assemble the properties to show the number of rounds to reach an $(1+\varepsilon)$-approximate market equilibrium is polynomially bounded.

Lemma 2.3. *For any market that satisfies Assumptions 2.1 and 2.2, and for any $\varepsilon > 0$, Alg-WGS-Precise returns the price vector of an $(1+\varepsilon)$-approximate market equilibrium in a number of rounds polynomial in the input size and $\log 1/\varepsilon$.*

Proof. Let x_t be the value of x we find in round t of Alg-WGS-Precise. First, because at least one price will increase by a factor of x_t in round t, by Claim 2.3 we have $\prod_t x_t \leq 2^{mD_1}$. At the end of round t, if $x_t \geq 1 + \frac{1}{Rm^3}$, let $s = \max\{s_i \mid s_i \in \mathcal{O}(\mathbf{p}_{t-1})\}$ and $s' = \max\{s_i \mid s_i \in \mathcal{O}(\mathbf{p}_t)\}$, then by Claim 2.4 we have $\Phi(\mathbf{p}_t) \leq ms'^2 \leq mx_t^2s^2 \leq mx_t^2\Phi(\mathbf{p}_{t-1})$. Moreover, by Lemma 2.2, there will be at most $O(m^4D_1)$ such rounds. Hence, the total increase of $\Phi(\mathbf{p}_t)$ in these rounds will be no more than a factor of $\prod_{x_t \geq 1+1/Rm^3} mx_t^2 \leq m^{O(m^4D_1)}2^{2mD_1} = m^{O(m^4D_1)}$.

For all other rounds, we have $x < 1 + \frac{1}{Rm^3}$, and by Lemma 2.1, the potential function is decreased by a factor of $1/(1 - \Omega(\frac{1}{m^3}))$. Therefore, the total number of rounds before $\Phi(\mathbf{p}_t) \leq \varepsilon^2$ will be at most

$$O\left(\log_{1/(1-\Omega(\frac{1}{m^3}))} \frac{m^{O(m^4D_1)}}{\varepsilon^2}\right) = O\left(D_1m^7\log m + m^3\log\frac{1}{\varepsilon}\right). \quad □$$

The final step is to argue that the total running time (in terms of oracle queries) of the algorithm is polynomial. Given Lemma 2.3, it remains to show that each round only invokes polynomially many oracle queries. Intuitively, this can be achieved by searching for x in Alg-WGS-Precise via binary search. However, for a formal proof, it is unavoidable to make statements about the required bit precision of $\mathbf{p}$, x, and the approximate demand oracle, which we did not discuss here. In the following, we present the statement of the main theorem. The formal proof, together with the statement of algorithm Alg-WGS based on bounded precision and full details on its analysis, is deferred to Appendix A.

Theorem 2.4. *For any market that satisfies Assumptions 2.1 and 2.2, and for any $\varepsilon > 0$, Alg-WGS returns the price vector of an $(1 + \varepsilon)$-approximate market equilibrium in time polynomial in the input size and $\log 1/\varepsilon$.*

Remark 2.1. Our main goal in the analysis was to establish a bound on the running time that is polynomial in m, L, and $\log 1/\varepsilon$. For the sake of simplicity, we did not optimize the bounds beyond being polynomial. It appears that the dependence on m can be significantly improved, for example, by a more precise analysis of the actual number of rounds with $x \geq 1 + \frac{1}{Rm^3}$ and their impact on the potential. Moreover, based on our preliminary experiments, it appears that instead of the precision parameter $\mu = \Theta(\varepsilon/m^7)$ (for details, see Appendix A), a value of $\Theta(\varepsilon/m^4)$ is sufficient, and the algorithm converges to an equilibrium much faster than the bound predicts.

3 EXCHANGE MARKETS WITH SPENDING CONSTRAINT UTILITIES

In this section, we discuss our algorithm for exchange markets with spending constraint utilities. Spending constraint utilities are defined in Devanur and Vazirani [24] and Vazirani [49], where the utility derived by agent i from good j is given by a PLC function f_{ij}. The overall utility of agent i is additively separable among goods (i.e., $u_i(\mathbf{x}) = \sum_{j \in G} f_{ij}(x_{ij})$). Each f_{ij} is a PLC function with a number of linear segments. Each segment k has two parameters: the rate of utility u_{ijk} per unit of good derived on segment k and the maximum fraction B_{ijk} of budget that can be spent on segment k. All B_{ijk} are strictly positive, and concavity implies $u_{ijk} > u_{ij(k+1)}$. Here we assume all u_{ijk}'s are integers, all B_{ijk}, w_{ij}'s are rational numbers, and the whole input can be represented in no more than L bits.

Spending constraint markets may not have an equilibrium [21]; however, under mild conditions, there is always a rational equilibrium [42]. Henceforth, we will assume the following sufficient condition. Let $\Gamma(S) = \{j \in G \mid w_{ij} > 0, i \in S\}$.

Assumption 3.1 (Sufficiency Condition). *For any subset S of agents, if $\Gamma(S) \neq G$ then there exists $i \in S$ and $j \in G \setminus \Gamma(S)$ such that $u_{ij1} > 0$.*

Let us first characterize the demand of each agent i under spending constraint utilities. Given nonzero prices $\mathbf{p}$, define the bang -per buck relative to $\mathbf{p}$ for segment k in f_{ij} to be u_{ijk}/p_j. Sort all segments of agent i by decreasing bang-per-buck value and partition them into t classes $Q_1, Q_2, \ldots, Q_t$, such that segments in the same class have the same bang-per-buck value. Then an allocation is a demand bundle of agent i if and only if there is an integer t_i such that all segments in partitions $Q_1, \ldots, Q_{t_i-1}$ are all fully allocated and no segments in partitions $Q_{t_i+1}, Q_{t_i+2}, \ldots$ are allocated. Furthermore, the total money spent on partitions $Q_1, \ldots, Q_{t_i-1}$ is no more than agent i's total budget $m_i = \sum_{j \in G} p_j w_{ij}$. We term Q_{t_i} agent i's *current partition* and $Q_1, \ldots, Q_{t_i-1}$ agent i's *allocated partition.* Let $spent_i^a$ denote the total money of agent i spent on allocated segments, and let $spent_j^g$ denote the total money spent on allocated segments of good j. Agent i can freely demand any segments in her current partition, fully or partially, until her remaining budget $m_i - spent_i^a$ is exhausted.

Equality graph. The main tool for the analysis of previous algorithms in spending constraint markets is an *equality graph*, denoted by $EG(\mathbf{p})$. This graph remains completely unknown to our algorithm, but it proves useful when proving properties of the convergence process. The vertex set of this bipartite graph consists of the set of agents A and the set of goods G. Given a price vector $\mathbf{p}_t$, we introduce an edge from agent i to good j if and only if agent i's current partition Q_{t_i} contains one segment that belongs to utility function f_{ij} for good j. The edges in $EG(\mathbf{p})$ are called *equality edges*. Observe that this graph changes throughout the process when we update the price vector $\mathbf{p}_t$.

Based on $EG(\mathbf{p})$, we can construct an *equality network* denoted by $N(\mathbf{p})$: First, for each edge in $EG(\mathbf{p})$ from agent i to good j, let k be the corresponding segment for f_{ij} that belongs to Q_{t_i}. We assign a capacity of $c_{ij} = B_{ijk}m_i$ to this edge. Next, add a source vertex s and a sink vertex t. For each agent i, add an edge from s to i with capacity $m_i - spent_i^a$, and finally add an edge from every good to sink t with infinite capacity. It is easy to see that every maximum flow in $N(\mathbf{p})$ corresponds to a feasible demand allocation for each agent.

Similar to Devanur et al. [23] and Vazirani [49], we define a *balanced flow* as a maximum flow in $N(\mathbf{p})$ that minimizes $\sum_{j \in G}(\ell_{jt} + spent_j^g - p_j)^2$, where ℓ_{jt} is the flow along edge (j, t), which also denotes the money spent on good j on segments of current partitions, and $spent_j^g$ is the amount of money spent on allocated partitions of good j. By assumption, $\mathcal{O}(\mathbf{p})$ returns the surplus vector derived from (any) balanced flow of the network $N(\mathbf{p})$. From Vazirani [49], we know that every balanced flow gives the same surplus vector, and such surplus vector can be computed using at most n max-flow computations.

There exists an algorithm for computing exact equilibrium prices in Fisher markets with spending constraint utilities [49]. Devanur and Vazirani [24] give an FPTAS for exchange markets with spending constraint utilities. The algorithm finds an $(1 + \varepsilon)$-approximate equilibrium in time polynomial in the input size and $1/\varepsilon$.

For spending constraint markets, we extend our approach for WGS markets. The challenge is that surplus can change in a non-continuous way when prices change the current partitions of the agents. However, we show how to use the linear structure of the market to get rough information about these breakpoints. In addition, we maintain prices within a polynomial precision and guarantee convergence to an approximate market equilibrium. Finally, when $\Phi(\mathbf{p})$ becomes small enough, we convert the approximate equilibrium to an exact one using a procedure Alg-Spending-Exact.

Our only assumption is that the whole input can be represented within L bits, and L is known to the algorithm. This implies as a corollary a variant of Assumption 2.1—there is an exact market equilibrium with prices $\mathbf{p}$ subject to $\frac{\max_i p_i}{\min_i p_i} \leq 2^{D_1}$, where D_1 is a polynomial of m and L. As mentioned before, Assumption 2.2 does not hold in spending constraint markets.

3.1 The Framework

Alg-Spending specifies the general framework of our algorithm, which is similar to the approach taken previously for WGS markets but with two notable differences at lines 8 and 10. As mentioned in Section 1.1, here we do not resort to approximation parameter μ, but instead compute the exact surplus. We first analyze Alg-Spending and show that it needs only a polynomial number of rounds to converge to an approximate market equilibrium. For now, our analysis disregards all precision and representation issues. In particular, we assume to find the exact value x using binary search, irrespective of the number of bits needed for representation. In addition, the update of prices from $\mathbf{p}_{t-1}$ to $\mathbf{p}_t$ will multiply all prices of goods in G_1 by x, irrespective of the number of bits required to represent them. In our final algorithm in the following, we will show how to address these issues to obtain a (true) polynomial-time algorithm.

The analysis of Alg-Spending proceeds roughly as in the previous section. We rely on the following lemma.

ALGORITHM 2: ALG-SPENDING: Framework for Spending Constraint Markets

Input: number of goods m, demand oracle O, number of bits L to represent the whole input (including each u_{ijk}, B_{ijk}, w_{ij}); solution precision ε
Output: Prices $\mathbf{p}$ of a $(1+\varepsilon)$-approximate market equilibrium
Parameters: $\varepsilon' = \frac{\varepsilon}{2\sqrt{m}}$

Set initial price $\mathbf{p}_0 \leftarrow (1, 1, \ldots, 1)$ and round index $t \leftarrow 0$.
repeat //round t
  $t \leftarrow t + 1$
  $\mathbf{s} = (s_1, \ldots, s_m) \leftarrow O(\mathbf{p}_{t-1})$
  Sort $\mathbf{s}$ such that $s_{i_1} \geq s_{i_2} \geq \cdots \geq s_{i_m}$.
  Find smallest k such that $s_{i_{k+1}} \leq 0$ or $s_{i_k} > (1 + \frac{1}{m})s_{i_{k+1}}$.
  Set $G_1 \leftarrow \{i_1, \ldots, i_k\}$ and $G_2 \leftarrow [m] \backslash G_1$
  Binary search the smallest $x \in (1, \infty)$ such that in $\mathbf{s}' = O(\text{UPDATE}(\mathbf{p}_{t-1}, x, G_1))$, it holds $\min\{s'_i \mid i \in G_1\} \leq \max(\{s'_i \mid i \in G_2\} \cup \{0\})$.
  $\mathbf{p}_t \leftarrow \text{UPDATE}(\mathbf{p}_{t-1}, x, G_1)$
until $\|O(\mathbf{p}_t)\|^2 < \varepsilon'^2$
return $\mathbf{p}_t$

LEMMA 3.1. *The demands returned by $O(\mathbf{p})$ satisfy the WGS property.*

PROOF. We make use of a max-min fair property for balanced flows in linear markets proved in previous work. A vector $\mathbf{s}$ is called *max-min fair* if and only if for every feasible vector $\mathbf{s}'$ and i such that $s_i < s'_i$, there is some j with $s_j < s_i$ such that $s_j > s'_j$. The following claim is proved by Devanur et al. [23]. □

CLAIM 3.1. *[23] The surplus vector of a balanced flow in $N(G)$ is max-min fair among all feasible surplus vectors.*

Although this claim is for linear markets, it can also be directly applied to spending constraint markets because the network flows are designed only with respect to the current partition of each agent i. Within this domain, the spending constraint market behaves exactly like a linear market.

We proceed to prove Lemma 3.1 by contradiction to this claim. Suppose we increase the price of some good k from p_k to $p_k + \delta$. We denote the old prices by $\mathbf{p}$ and the new prices by $\mathbf{p}'$. Let $\mathbf{s} = O(\mathbf{p})$ and $\mathbf{s}' = O(\mathbf{p}')$. Now assume for contradiction that there exists some $\ell \neq k$ such that $s'_\ell < s_\ell$. Let $S_G = \{j \in G \mid s_j \geq s_\ell, j \neq k\}$ and $S_A = \{i \in A \mid \text{there exists } j \in S_G \text{ such that } f_{ij} > 0 \text{ in } N(\mathbf{p})\}$. It is easy to verify the following properties:

(a) For any edge (i, j) in $N(\mathbf{p})$ with $j \neq k$, let *seg* be the segment in agent i's current partition that corresponds to this edge. Then with new prices $\mathbf{p}'$, either *seg* belongs to the allocated partition (i.e., is fully allocated) or *seg* is still in agent i's current partition (i.e., (i, j) also exists in $N(\mathbf{p}')$).
(b) For any edge $(i, j) \in N(\mathbf{p}')$ with $i \in S_A$ and $j \notin S_G$, let *seg* be the segment in agent i's current partition that corresponds to this edge with prices $\mathbf{p}'$. If *seg* is also in i's current partition with prices $\mathbf{p}$, then *seg* is fully allocated with price $\mathbf{p}$ (otherwise, agent i can reroute some flow from S_G to this segment to obtain a more balanced flow[7] in $N(\mathbf{p})$). If *seg* is not in i's current partition with price $\mathbf{p}$, then all segments in i's current partition with price $\mathbf{p}$ must be fully allocated in the demand allocation with prices $\mathbf{p}'$.

[7]Here "more balanced" means the new flow has a smaller value of $\sum_{j \in G}(\ell_{jt} + spent_j^g - p_j)^2$.

The preceding two observations imply that with prices $\mathbf{p}'$, we can rearrange flows from S_A to S_G to get a new feasible surplus vector $\mathbf{s}''$ such that for all $j \in S_G$, $s''_j \geq s_j$, and for all other edges $j \notin S_G$, $s''_j = s'_j$. In particular, we have $s''_j \geq s_j \geq s_\ell > s'_\ell$ for all $j \in S_G$. This contradicts the fact that $\mathbf{s}'$ is max-min fair and proves Lemma 3.1.

Now the following properties can be proved using literally the same proofs as for WGS markets before. Again we set the constant $R = 48e^2$.

Claim 3.2. *For Alg-Spending, the following properties hold.*

(1) *In Alg-Spending, $|O(\mathbf{p}_0)| \leq 2m$, and $|O(\mathbf{p}_t)|$ is non-increasing in t.*
(2) *$O(\text{Update}(\mathbf{p}, x, S))_i \leq x \cdot O(\mathbf{p})_i$, for any price vector $\mathbf{p}$, $x > 1$, $S \subseteq [m]$, and $i \in S$.*
(3) *The number of rounds that end with $x \geq 1 + \frac{1}{Rm^3}$ in Alg-Spending is $O(m^4 D_1)$.*

We also show a version of Claim 2.2 for spending constraint markets, which needs some extra work. Unlike for WGS markets, the surplus of a good can change from non-negative to negative. Thus, the proof of Claim 2.2 does not directly transfer to spending constraint markets. Instead, we first show the following.

Claim 3.3. *Let S_t be the set of goods with negative surplus and price strictly greater than 1 at the end of round t in Alg-Spending. For any $T \subseteq S_t$, let $\Gamma(T, \mathbf{p}_t)$ be the neighbors of set T in $EG(\mathbf{p}_t)$—for instance, $\Gamma(T, \mathbf{p}_t)$ is the set of agents who are interested in at least one good in T under price $\mathbf{p}_t$. Let $B(\Gamma(T, \mathbf{p}_t))$ be the sum of budgets of agents in $\Gamma(T, \mathbf{p}_t)$. Then we have $B(\Gamma(T, \mathbf{p}_t)) > \sum_{i \in T} (\mathbf{p}_t)_i$.*

Proof. We prove this claim by induction. The claim is trivially true for round 0. Assume that it is true for any round $t \leq t'$, then at the end of round $t = t' + 1$, consider two cases:

- $\min\{s'_i \mid i \in G_1\} \geq 0$. Because the surplus of any good in G_2 is non-decreasing in round t, we have $S_t \subseteq S_{t-1}$. Further, the algorithm does not increase the price of any good in T in round t. Hence, we also have $\Gamma(T, \mathbf{p}_{t-1}) \subseteq \Gamma(T, \mathbf{p}_t)$. By the induction assumption, we have $B(\Gamma(T, \mathbf{p}_t)) \geq B(\Gamma(T, \mathbf{p}_{t-1})) > \sum_{i \in T} (\mathbf{p}_{t-1})_i = \sum_{i \in T} (\mathbf{p}_t)_i$.
- $\min\{s'_i \mid i \in G_1\} < 0$. Assume that during this round we start from $x = 1$ and increase x continuously until it reaches its final value. We also assume that the equality graph $EG(\text{Update}(\mathbf{p}_{t-1}, x, G_1))$ and the corresponding balanced flow are implicitly being maintained throughout the process. For each agent i and any moment during this round, let $\Gamma(i)$ denote the neighbors of agent i in the equality graph. There are two cases:

  Case (a):. $\Gamma(i) \cap G_1 \neq \emptyset$ and $\Gamma(i) \cap G_2 \neq \emptyset$. This means agent i's current partition contains segments of goods in both G_1 and G_2. When we continue to increase x (and consequently the prices of goods in G_1), the segments of goods in G_1 will have worse bang-per-buck value than segments of goods in G_2. Hence, they will be removed from agent i's current partition. Furthermore, for any good $j_1 \in \Gamma(i) \cap G_1$ and $j_2 \in \Gamma(i) \cap G_2$, because we have $s_{j_1} > s_{j_2}$ and the balanced flow condition, it cannot happen that $\ell_{ij_1} > 0$ and $\ell_{ij_2} < c_{ij_2}$. Otherwise, agent i would be able to route some flow from edge (i, j_1) to edge (i, j_2) to reach a more balanced flow. Thus, two possibilities remain:

  - $\ell_{ij_2} = c_{ij_2}$ for every $j_2 \in \Gamma(i) \cap G_2$. This means that all segments of goods in G_2 are already fully allocated. When x increases, the new current partition of agent i will only contain segments of goods in $\Gamma(i) \cap G_1$, and the surpluses will change continuously with the change of x.
  - $\ell_{ij_1} = 0$ for every $j_1 \in \Gamma(i) \cap G_1$. This means that when x increases, the segments that are being removed from i's current partition are all unallocated in the current allocation. Hence, the surpluses will again change continuously at the current point.

In summary, in this case, the change of the surpluses will always be continuous in the change of x. Note that when the surpluses are changing continuously, Case 2 cannot occur before Case 1 happens. Therefore, the claim follows via Case 1.

Case (b):. $\Gamma(i) \subseteq G_1$ or $\Gamma(i) \subseteq G_2$. This means that we will always change the prices of all segments in i's current partition by the same rate. Hence, if set Q_{t_i} changes, it can only be merged with some segments in Q_{t_i-1} or Q_{t_i+1}. In either case, the final value of x must be at a point where at least one new edge emerges in the equality graph $EG(\mathbf{p}_t)$ from $\Gamma(G_1) \subseteq A$ (the set of agents incident to at least one good in G_1) to G_2. Suppose we alter the equality graph by removing these emerging edges from $EG(\mathbf{p}_t)$. If we recompute the balanced flow, then $\min\{s'_i \mid i \in G_1\} > \max\{\{s'_i \mid i \in G_2\} \cup \{0\}\}$. For any $T \subseteq S_t$, let $T_1 = T \cap G_1$ and $T_2 = T \cap G_2$. In this new graph, let $\Gamma'(T_1)$ be the set of agents who have positive flow to at least one good in T_1, and let $\Gamma'(T_2)$ be the set of agents incident to at least one good in T_2. Since $\min\{s'_i \mid i \in T_1\} > \max\{s'_i \mid i \in T_2\}$, by the balanced flow condition we know $\Gamma'(T_1) \cap \Gamma'(T_2) = \emptyset$. In addition, we have $B(\Gamma'(T_1)) > \sum_{i\in T_1}(\mathbf{p}_t)_i$ because every good in T_1 has positive surplus, and $B(\Gamma'(T_2)) > \sum_{i\in T_2}(\mathbf{p}_t)_i$ because the claim is true in round $t-1$. Combining these two inequalities gives us $B(\Gamma(T, \mathbf{p}_t)) > \sum_{i\in T}(\mathbf{p}_t)_i$. □

Corollary 3.2. *At the end of each round t in Alg-Spending, for any good i with negative surplus and price greater than 1, there exists another good j with price 1 that is connected to i in $EG(\mathbf{p}_t)$*

Proof. Assume for contradiction that the statement is false. Let T be the set of goods with negative surplus and connected with good i in $EG(\mathbf{p}_t)$. By the balanced flow condition, none of the agents in $\Gamma(T, \mathbf{p}_t)$ can have any positive flow to goods outside set T. Thus, we have $0 > \sum_{i\in T} s'_i = B(\Gamma(T, \mathbf{p}_t)) - \sum_{i\in T}(\mathbf{p}_t)_i$. This contradicts Claim 3.3. □

We obtain the following corollary, an analog of Claim 2.2 for spending constraint markets.

Corollary 3.3. *Throughout the run of Alg-Spending, there will be at least one good whose price remains 1.*

The set of properties shown so far allows to establish the following lemma, which is the key step for observing convergence to equilibrium. It can be seen as an adjustment of Lemma 2.1 to spending constraint markets.

Lemma 3.4. *If $x < (1 + \frac{1}{Rm^3})$ at the end of round t in Alg-Spending, then $\Phi(\mathbf{p}_t) \leq \Phi(\mathbf{p}_{t-1})(1 - \frac{1}{16e^2m^3})$.*

Proof. Our proof uses the arguments of the proof for Lemma 2.1. We consider two cases:

Case 1: $\min\{s'_i \mid i \in G_1\} = \max\{\{s'_i \mid i \in G_2\} \cup \{0\}\}$. This case can be verified by observing that Claim A.4 holds with $\mu = 0$. Then the proof follows using exactly the same proof as for Lemma 2.1.

Case 2: $\min\{s'_i \mid i \in G_1\} < \max\{\{s'_i \mid i \in G_2\} \cup \{0\}\}$. Using the same argument as in the proof of Claim 3.3, one can show that in this case x must be at a point where at least one new edge emerges in the equality graph $EG(\mathbf{p}_t)$ from $\Gamma(G_1) \subseteq A$. Without these emerging edges, we have $\min\{s'_i \mid i \in G_1\} \geq \max\{s'_i \mid i \in G_2\}$. This means that we can further reduce the flows along the new edges to get another feasible flow in $N(\mathbf{p}_t)$ such that with the resulting surplus vector s'', we have $\min\{s''_i \mid i \in G_1\} = \max\{s''_i \mid i \in G_2\}$.

Next, using again the same proof as for Lemma 2.1, for our choice of the value of constant R it follows that $\|s''\|^2 \leq \Phi(\mathbf{p}_{t-1})(1 - \frac{1}{16e^2m^3})$. Hence, $\Phi(\mathbf{p}_t) \leq \|s''\|^2 \leq \Phi(\mathbf{p}_{t-1})(1 - \frac{1}{16e^2m^3})$. □

ALGORITHM 3: Rounding $(\mathbf{p}, M)$: Rounding Procedure

Input: price vector $\mathbf{p}$ in which $\min_i \mathbf{p}_i = 1$, rounding bound $M \geq L$

Output: Rounded price vector $\mathbf{p}'$

Let $\mathcal{P} = \{\frac{a}{b} \mid a, b \in \mathbb{Z}^+, a, b \leq 2^M\}$ and $\mathbf{p}' \leftarrow \mathbf{p}$.
while *$RG(M, \mathbf{p}')$ is not connected* **do**
    Let $C_1, C_2, \ldots, C_k$ be the connected components of $RG(M, \mathbf{p}')$
    Assume without loss of generality that C_1 is a component with $\min_{i \in C_1 \cap A} p'_i = 1$
    For every i, j, let $r_{ij} = p'_i / p'_j$.
    Let $h_{ij} = \min\{\frac{x}{r_{ij}} \mid x \in \mathcal{P}, x \geq r_{ij}\}$ and $H_{ij} = \min\{h_{i'j'} \mid i' \in C_i, j' \in C_j\}$.
    Let $(i, j) \in \arg\min_{i > j} H_{ij}$.
    $\mathbf{p}' \leftarrow$ Update$(\mathbf{p}', H_{ij}, C_i)$
end
return $\mathbf{p}'$

3.2 Precision and Representation

It is now tempting to think that using a similar argument as in the general WGS case, Claim 3.2, Corollary 3.3, and Lemma 3.4 provide an algorithm that converges to an approximate market equilibrium in polynomial time. However, an issue arises with regard to the precision and representation of the prices: In each round, x could potentially be a rational number involving prices and surpluses, and after multiplying each price in G_1 by x, the bit length to represent a price can double in one round. This means that after a polynomial number of rounds, we may require an exponential number of bits to represent the prices of some goods and the desired factor x.

Ratio graph.

Definition 3.5. The *ratio graph* $RG(M, \mathbf{p})$ is an undirected graph with m vertices (where m is the number of goods in the market), and for any two goods i and j, (i, j) is an edge if and only if p_i/p_j can be represented as a ratio of two integers, each of value at most $2^M - 1$.

The reason for defining such ratio graph is that, although the equality subgraph with respect to a price vector is unknown to us, we can use the ratio graph, which can be computed using only the price vector and the input size bound, to retrieve some information about the hidden structure of the equality subgraph.

Claim 3.4. *Let L be the upper bound on the number of bits to represent each utility parameter and $\mathbf{p}$ be a price vector. For any price vector $\mathbf{p}$ and goods i, j that are connected in $EG(\mathbf{p})$, i and j are also connected in $RG(M, \mathbf{p})$ for any $M \geq L$.*

Proof. If good i and good j are connected in $EG(\mathbf{p})$, then there exist goods $i = i_0, i_1, \ldots, i_{k-1}, i_k = j$ such that for each $t < k$, there exists some agent a_t that has the same bang per buck for two segments for goods i_t and i_{t+1}, respectively. Then we have $p_{i_t}/p_{i_{t+1}} = u_{a_t i_t k_1}/u_{a_t i_{t+1} k_2}$ for some k_1 and k_2. This is a ratio of two integers, each of value at most $2^L - 1$. This implies $(i_t, i_{t+1}) \in RG(M, \mathbf{p})$, so i and j are connected in $RG(M, \mathbf{p})$. □

As an adjustment, we run the following procedure Rounding$(\mathbf{p}, M)$ at the end of each iteration of the main loop in Alg-Spending. Its purpose is to round the prices within polynomial bit length while maintaining the structure of equality and ratio graphs. Thereby, we will show that the value of potential function $\Phi(\mathbf{p})$ will not be increased dramatically.

Our new algorithm Alg-Spending-Rounding is simply the framework Alg-Spending with the following modifications:

(1) Set $M = \log_2 \frac{5m^7}{\varepsilon'^2}$. In each round, binary search x within domain $\mathcal{P} = \{\frac{a}{b} \mid a > b, a, b \in \mathbb{Z}^+, a, b \leq 2^{2mM+L}\}$ instead of $(1, \infty)$.
(2) At the end of each round t, update $\mathbf{p}_t \leftarrow$ Rounding$(\mathbf{p}_t, M)$ with M as earlier.

Lemma 3.6. *Given any price vector* $\mathbf{p}$, *Rounding*$(\mathbf{p}, M)$ *terminates in time polynomial in* m, M *and the bit length to represent* $\mathbf{p}$. *The returned price vector* $\mathbf{p}'$ *satisfies:*

(a) there exists $i \in G$ *with* $p_i = p_i' = 1$,
(b) every price can be represented as a ratio of two integers, each of value at most 2^{mM},
(c) $EG(\mathbf{p}')$ *contains every edge present in* $EG(\mathbf{p})$, *and*
(d) $p_i \leq p_i' \leq p_i + 2^{-M}$ *for every* $i \in G$.

Proof. Property (a) holds since we never increase the price of goods in set C_1, and property (b) can be derived based on property (a) and the fact that $RG(M, \mathbf{p}')$ is connected when the algorithm terminates.

Next we claim that for any i, j, there does not exist any $x \in \mathcal{P}$ such that $p_i/p_j \leq x < p_i'/p_j'$. This is because by design, p_i/p_j has to reach x in some iteration before it grows beyond x. But starting from that moment until the end of the algorithm, i and j will be connected in $RG(M, \mathbf{p})$. The ratios between two prices in the same connected component in $RG(M, \mathbf{p})$ remain unchanged. Hence, p_i/p_j will never grow beyond x.

This claim also proves property (c). By Claim 3.4, a pair of goods i and j connected in $EG(\mathbf{p})$ remain connected in $RG(M, \mathbf{p})$. Hence, their ratio of prices will remain the same in $\mathbf{p}'$, and they are connected in $EG(\mathbf{p}')$.

For property (d), according to the algorithm, no price will decrease from $\mathbf{p}$ to $\mathbf{p}'$. Next, number the goods such that $p_1 = p_1' = 1$. For any $i \neq 1$, let $x_i = \min\{x \in \mathcal{P} \mid x \geq p_i\}$. Then we have $p_i'/p_1' = p_i' \leq x_i$, as well as $x_i \leq p_i + 2^{-M}$. Therefore, $p_i \leq p_i' \leq p_i + 2^{-M}$.

Finally, in each iteration, we add at least one edge between two connected components in $RG(M, \mathbf{p})$. Thus, the algorithm will terminate after at most $m - 1$ iterations, and it is easy to check that each iteration runs in polynomial time. This proves the claim. □

Once the prices are bounded by a fixed polynomial bit length, we can also bound the length needed to encode the desired x in each round. This implies that we can find x in the framework in polynomial time using binary search.

Lemma 3.7. *In every round of Alg-Spending-Rounding, the desired* x *can be represented as a ratio of two integers, each of value at most* $2^{mM+L+2\log m} - 1$.

Proof. Let $\mathbf{p} = \mathbf{p}_{t-1}$ and consider the structure of $EG(\mathbf{p})$. We let $A_1 = \Gamma(G_1)$ be the agents connected to goods in G_1. In addition, let $A_2 = A \setminus A_1$. There is no $(i, j) \in EG(\mathbf{p})$ with $i \in A_1$ and $j \in G_2$, as otherwise $\mathcal{O}(\mathbf{p}_{t-1})$ could increase the money spent on goods in G_2 and further decrease $\Phi(\mathbf{p}_{t-1})$. For simplicity, we will also assume that there is no edge $(i, j) \in A_2 \times G_1$, as no agent spends money along these edges and they immediately disappear once we start increasing prices in G_1.

Now we increase $\mathbf{p}$ on goods in $j \in G_1$ by x and get a new price vector $\mathbf{p}(x)$. This only generates new edges $(i, j) \in A_1 \times G_2$. Furthermore, we drop only edges $(i, j) \in A_1 \times G_1$. To verify this, let us consider the other possibilities. The relation between marginal utility values u_{ijk}/p_j and $u_{ij'k'}/p_{j'}$ for goods in $j, j' \in G_1$ does not change, as both p_j and p_k are both multiplied by x. Hence, there are no new edges $(i, j) \in A_1 \times G_1$. For the same reason, there are no new edges $(i, j) \in A_2 \times G_2$. The bang per buck of goods in G_1 decreases, so we also do not introduce edges $(i, j) \in A_2 \times G_1$. In fact, this also implies that we do not drop any edges $(i, j) \in A_2 \times G_2$—prices and bang-per-buck relations among goods in G_2 do not change at all, and G_2 becomes more attractive compared to

G_1. Finally, there exist no edges $(i,j) \in (A_1 \times G_2) \cup (A_2 \times G_1)$ that could be removed. This shows that we only generate new edges between A_1 and G_2, and we only drop edges between A_1 and G_1.

For any given x, consider the residual graph of $N(\textsc{Update}(\mathbf{p}, x, G_1))$. Let C be an arbitrary connected component in this graph. Let C_A be the set of agents in C and C_G be the set of goods in C. Then we know that all goods in C_G have the same surplus, and all flow going through C_G comes from agents in C_A. This implies the following equation:

$$\sum_{i \in C_A \setminus G_1} p_i + x \sum_{i \in C_A \cap G_1} p_i = \sum_{i \in C_G \setminus G_1} p_i + x \sum_{i \in C_G \cap G_1} p_i + |C_G| s,$$

where s is the surplus of (any) good in this component.

Now let us focus on the moment where x reaches the desired value at the end of round t according to the algorithm. At this moment, one of the following properties must hold:

(1) $\min\{s'_i \mid i \in G_1\} = 0$. Then for the connected component that contains a good of surplus 0, we have $s = 0$ in the preceding equation. All initial prices p_i are ratios of integers with values at most 2^{mM}, so when we solve the equation for x, the solution is a ratio of two integers with value at most $m2^{mM} = 2^{mM+\log m}$.

(2) $\min\{s'_i \mid i \in G_1\} = \max\{s'_j \mid j \in G_2\}$. In this case, we have two possibilities:
- There exist two connected components in the residual graph of $N(\textsc{Update}(\mathbf{p}, x, G_1))$ that have the same surplus. Applying the preceding equation to these two components, we can solve for x, and the solution will be a ratio of two integers with value at most $m^2 2^{mM} = 2^{mM+2\log m}$.
- A new edge $(i,j) \in A_1 \times G_2$ appears, then for agent i good $j \in G_2$ becomes equally attractive as some $k \in G_1$: $u_{ij}/p_j x = u_{ik}/p_k$, or, equivalently, $x = (u_{ij} p_k)/(u_{ik} p_j)$. p_{ij} and p_k can be represented as ratio of integers of value at most 2^{mM} by Lemma 3.6, and u_{ij} and u_{ik} are both integers of value at most $2^L - 1$. Hence, every value of x at which a new edge evolves in $EG(\mathbf{p}(x))$ can be presented as a ratio of integers of value at most 2^{2mM+L}. □

We now bound the impact of replacing $\mathbf{p}_t$ by $\textsc{Rounding}(\mathbf{p}_t, M)$ in the function $\Phi(\mathbf{p}_t)$.

Lemma 3.8. *$\Phi(\textsc{Rounding}(\mathbf{p}_t, M)) < \Phi(\mathbf{p}_t) + 5m^3 2^{-M}$ for any round t.*

Proof. Let $\mathbf{p}'_t = \textsc{Rounding}(\mathbf{p}_t, M)$. By Lemma 3.6(c), we know this rounding procedure does not remove any edges in $EG(\mathbf{p}_t)$. Let f be a balanced flow in $N(\mathbf{p}_t)$. Then by Lemma 3.6(d), we can construct a feasible flow f' in $N(\textsc{Rounding}(\mathbf{p}_t, M))$ such that $f_{ij} \le f'_{ij} \le f_{ij} + 2^{-M}$ for every i, j. Let $\mathbf{s}$ be the surplus vector derived from f', then we have $|s_i - O(\mathbf{p}_t)_i| < m2^{-M}$ for every i. Hence,

$$\begin{aligned} \|\mathbf{s}\|_2^2 - \|O(\mathbf{p}_t)\|_2^2 &= \sum_i (s_i^2 - O(\mathbf{p}_t)_i^2) \\ &\le \sum_i (2m2^{-M} |O(\mathbf{p}_t)_i| + m^2 2^{-2M}) \\ &= 2m^2 2^{-M} |O(\mathbf{p}_t)| + m^3 2^{-2M} \\ &\le 4m^3 2^{-M} + m^3 2^{-2M} \\ &< 5m^3 2^{-M}. \end{aligned}$$

Note that $\mathbf{s}$ is just one feasible surplus vector for price vector $\mathbf{p}'_t$, and $\Phi(\mathbf{p}'_t)$ minimizes the ℓ_2-norm of surpluses among all feasible surplus vectors. Hence, $\Phi(\textsc{Rounding}(\mathbf{p}_t, M)) \le \|\mathbf{s}\|_2^2 < \|O(\mathbf{p}_t)\|_2^2 + 5m^3 2^{-M}$. This proves the lemma. □

ROUNDING can be used to obtain an algorithm that converges to an approximate market equilibrium in polynomial time.

LEMMA 3.9. *For every spending constraint exchange market satisfying Assumption 3.1, an* $(1+\varepsilon)$*-approximate market equilibrium can be computed in time polynomial in m, L, and* $\log 1/\varepsilon$.

PROOF. In the ALG-SPENDING framework with ALG-SPENDING-ROUNDING, we know by Lemma 3.4 that at the end of each round t and before calling ROUNDING, $\Phi(\mathbf{p}_t) \le \Phi(\mathbf{p}_{t-1})(1 - \Omega(\frac{1}{m^3}))$. If $\Phi(\mathbf{p}_t) > \varepsilon'^2$, we have $5m^3 2^{-M} = \varepsilon'^2/m^4 < \Phi(\mathbf{p}_t)/m^4$. Thus, by Lemma 3.8,

$$\begin{aligned}\Phi(\text{ROUNDING}(\mathbf{p}_t, M)) &\le \left(1 + \frac{1}{m^4}\right)\Phi(\mathbf{p}_t)\\ &\le \left(1 + \frac{1}{m^4}\right)\left(1 - \Omega\left(\frac{1}{m^3}\right)\right)\Phi(\mathbf{p}_{t-1})\\ &= \left(1 - \Omega\left(\frac{1}{m^3}\right)\right)\Phi(\mathbf{p}_{t-1}).\end{aligned}$$

This implies that we can employ the same proof as for Theorem 2.4 to show that after finishing ALG-SPENDING-ROUNDING, we arrive at a $(1+\varepsilon)$-approximate market equilibrium. Because M is a polynomial in the input size and $\log 1/\varepsilon$, the binary search and the ROUNDING procedure run in polynomial time in each round of the framework. Hence, the running time is polynomial in the input size and $\log 1/\varepsilon$. □

Finally, it remains to convert the approximate equilibrium to an exact one. To achieve this, we rely on full information about the spending constraint utilities. Although this step can be seen as an extension of the technique developed in the work of Duan and Mehlhorn [26] for the linear exchange markets, there are several challenges due to the much more involved setting of spending constraint utilities, where the allocated partitions make the remaining budgets of agents and the values of goods dependent on too many parameters. Using a more involved procedure, we are able to handle the extra complexity of the problem. Our result resolves an open question of Duan and Mehlhorn [26] of finding an exact polynomial-time algorithm for exchange markets with spending constraint utilities. A detailed discussion of this final step can be found in Appendix B. This yields the final theorem in this section.

THEOREM 3.10. *For every spending constraint exchange market satisfying Assumption 3.1, ALG-SPENDING-EXACT returns the price vector of a market equilibrium in time polynomial in m and L.*

APPENDIX

A WGS EXCHANGE MARKETS WITH APPROXIMATE PRECISION

In this section, we describe the complete algorithm ALG-WGS for WGS exchange markets. Recall the assumptions from Section 2, which we restate here for completeness.

ASSUMPTION 2.1. *There exists a market equilibrium* $(\mathbf{p}^*, \mathbf{x}^*)$ *with* $1 \le p_i^* \le 2^{D_1}, \forall i \in G$.

ASSUMPTION 2.2. *For any price vector* $\mathbf{p}$ *such that* $1 \le p_i \le 2^{D_1}$ *for each i,* $|\frac{\partial s_i}{\partial p_j}| < 2^{D_2}$ *for every* i, j*, where* s_i *is the surplus money of good i in* $\mathcal{O}(\mathbf{p})$*, and* D_2 *is a polynomial of the input size.*

Throughout the analysis and proofs that follow, if $\mathbf{s} = \mathcal{O}(\mathbf{p})$ for some $\mathbf{p}$, we use $\tilde{\mathbf{s}}$ to denote the surplus vector returned by the μ-approximation demand oracle with the same price vector (i.e., $\tilde{\mathbf{s}} = \widetilde{\mathcal{O}}(\mathbf{p}, \mu)$). We proceed along similar lines as in Section 2, and the proofs of the first claims closely resemble the versions for the exact oracle. For completeness, we provide them here for the approximate oracle.

CLAIM A.1. *In ALG-WGS,* $|\mathcal{O}(\mathbf{p}_0)| \le 2m$ *and* $|\mathcal{O}(\mathbf{p}_t)|$ *is non-increasing in t.*

ALGORITHM 4: Alg-WGS

Input: number of goods m, approximate demand oracle $\widetilde{O}$, precision bound $\varepsilon > 0$
Output: Prices $\mathbf{p}$ of a $(1+\varepsilon)$-approximate market equilibrium
Parameters: $R_1 = 1728e^2$, $\mu = \frac{\varepsilon}{R_1 m^7}$, $\Delta = \frac{1}{\mu}(2^{D_1+D_2+\log m})$, $\varepsilon' = \frac{\varepsilon}{2\sqrt{m}}$

```
Set initial price p_0 ← (1, 1, ..., 1) and round index t ← 0.
Let 𝒫 = { a/b | a > b, a, b ∈ ℤ⁺, a, b ≤ Δ }
repeat //round t
    t ← t + 1
    s̃ = (s̃_1, ..., s̃_m) ← Õ(p_{t−1}, μ)
    Sort s̃ such that s̃_{i_1} ≥ s̃_{i_2} ≥ ··· ≥ s̃_{i_m}.
    Find smallest k such that s̃_{i_{k+1}} ≤ μ or s̃_{i_k} > (1 + 1/m) s̃_{i_{k+1}}.
    Set G_1 ← {i_1, ..., i_k} and G_2 ← G \ G_1
    Binary search the largest x ∈ 𝒫 such that in s̃' = Õ(Update(p_{t−1}, x, G_1)), it holds
        min{s̃'_i | i ∈ G_1} ≥ max{{s̃'_i | i ∈ G_2} ∪ {μ}}.
    p_t ← Update(p_{t−1}, x, G_1)
until ||Õ(p_t)||² < ε'²
return p_t
```

Proof. Let d_i be the exact demand for good i under price $\mathbf{p}_0$, then $|O(\mathbf{p}_0)| = \sum_i |d_i - 1| \leq \sum_i d_i + m = 2m$. Next, by the criteria to define G_1 and G_2 in each round, we have $\{i \mid O(\mathbf{p}_{t-1})_i < 0\} \subseteq G_2$: To see this, observe that the surplus resulting from the approximate $\widetilde{O}(\mathbf{p}_{t-1})$ differs by at most an additive $\mu = \varepsilon/(R_1 m^7)$, so a good i with $O(\mathbf{p}_{t-1})_i < 0$ will always be classified in G_2 with respect to $\widetilde{O}(\mathbf{p}_{t-1})_i$.

During round t, only prices of goods in G_1 are increased. By the WGS property, we know $O(\mathbf{p}_t)_i \geq O(\mathbf{p}_{t-1})_i$ for every $i \in G_2$. Further, note that $\min\{O(\mathbf{p}_t)_i \mid i \in G_1\} \geq 0$ since $\min\{\widetilde{O}(\mathbf{p}_t)_i \mid i \in G_1\} \geq \mu$. Hence, we do not introduce any new negative surplus in $O(\mathbf{p}_t)$. Thus, we have

$$|O(\mathbf{p}_{t-1})| = -2 \sum_{O(\mathbf{p}_{t-1})_i < 0} O(\mathbf{p}_{t-1})_i \geq -2 \sum_{O(\mathbf{p}_t)_i < 0} O(\mathbf{p}_{t-1})_i \geq -2 \sum_{O(\mathbf{p}_t)_i < 0} O(\mathbf{p}_t)_i = |O(\mathbf{p}_t)|. \qquad \square$$

The next two claims bound the range of prices we encounter, which is important for showing that we approach the unique market equilibrium.

Claim A.2. *Throughout the run of Alg-WGS, every good with negative surplus has price 1. Hence, there will be at least one good whose price remains 1.*

Proof. Observe the following three simple facts about the surplus $O(\mathbf{p}_t)$ resulting from exact demands. First, throughout the algorithm, we never increase the price of any good with negative surplus. Second, the surplus of any good does not change from non-negative to negative. Third, for any non-equilibrium price vector, there will always be a good with negative surplus. These facts are direct consequences of the conditions used to classify goods based on $\widetilde{O}(\mathbf{p}_t)$ in the algorithm. Together they prove the claim. □

Claim A.3. *In Alg-WGS, for any $t \geq 0$, all prices in $\mathbf{p}_t$ are bounded by 2^{D_1}.*

Proof. Let $\mathbf{p}^*$ be equilibrium prices according to Assumption 2.1. We show that for any $t \geq 0$, $\mathbf{p}_t$ is always pointwise smaller than $\mathbf{p}^*$. Assume that this is not true, and let t be the smallest value such that there exists $(\mathbf{p}_t)_i > \mathbf{p}^*_i$ for some i. Note that according to the algorithm, we have $\mathbf{p}_t = \text{Update}(\mathbf{p}_{t-1}, x, G_1)$ for some $x > 1$ and $G_1 \subseteq [m]$. Further, by our classification based on $\widetilde{O}$, it is easy to see that $\mathcal{O}(\mathbf{p}_t)_i > 0$ for any $i \in G_1$. This means from $\mathbf{p}_{t-1}$ to $\mathbf{p}_t$, only prices of goods in G_1 are increased. Let $S = \{i \mid (\mathbf{p}_t)_i > \mathbf{p}^*_i\}$, then we have $S \subseteq G_1$.

Next, we apply a sequence of price changes to $\mathbf{p}_t$. First, for every $j \notin S$, we increase $(\mathbf{p}_t)_j$ to $\mathbf{p}^*_j$. Let $\mathbf{p}$ be the new price vector, and consider the surplus $\mathcal{O}(\mathbf{p})$ resulting from exact demands. By the WGS property of the market, the surplus of any good in S will not decrease, and hence we still have $\mathcal{O}(\mathbf{p})_i > 0$ for every $i \in S$. The sum of all surpluses in an exchange market is always 0, so $\sum_{j \notin S} \mathcal{O}(\mathbf{p})_j < 0$.

Now, we decrease the price of every good $i \in S$ from $\mathbf{p}_i$ to $\mathbf{p}^*_i$. Then, $\mathbf{p}$ becomes exactly $\mathbf{p}^*$. This process will not increase surplus of any good $j \notin S$. Thus, we still have $\sum_{j \notin S} \mathcal{O}(\mathbf{p}^*)_j < 0$. This contradicts the assumption that $\mathbf{p}^*$ are prices of a market equilibrium. □

At this point, let us recall Claim 2.4 that establishes a relation between the exact surplus of a good before and after a multiplicative price update step. It does not involve the approximate oracle.

Next, we establish a statement about the surpluses at the end of each round, which was not necessary for the version with exact oracles and precision. Intuitively, we increase the prices of goods in G_1 until the minimum surplus in G_1 reaches the maximum surplus in G_2 or 0. Note that μ is very small and can be thought of as 0. The main complication here is that we need to work with μ-approximation demands in the algorithm and the resulting surpluses s'.

Claim A.4. *At the end of each round in Alg-WGS,* $\min\{\tilde{s}'_i \mid i \in G_1\} \le \max\{\{\tilde{s}'_i \mid i \in G_2\} \cup \{\mu\}\} + 6\mu$.

Proof. According to the binary search procedure, we know that if we increase prices in G_1 by a factor of x, then $\tilde{s}'$ satisfies the condition $\min\{\tilde{s}'_i \mid i \in G_1\} \ge \max\{\{\tilde{s}'_i \mid i \in G_2\} \cup \{\mu\}\}$. Furthermore, an increase by $x^+ = \min\{y \in \mathcal{P} \mid y > x\} < x + \frac{1}{\Delta}$ would result in a surplus vector that does not satisfy this condition. Let $\mathbf{s}^+ = \mathcal{O}(\text{Update}(\mathbf{p}_{t-1}, x^+, G_1))$. By Assumption 2.2, we have

$$|\tilde{s}^+_i - \tilde{s}'_i| \le |s^+_i - s'_i| + 2\mu < 2^{D_2} \cdot (x^+ - x)|\mathbf{p}_{t-1}| + 2\mu \le \frac{2^{D_2}|\mathbf{p}_{t-1}|}{\Delta} + 2\mu \le \frac{2^{D_2+D_1+\log m}}{\Delta} + 2\mu = 3\mu.$$

for every i, where the last inequality is derived by Claim A.3. Thus,

$$\begin{aligned}\min\{\tilde{s}'_i \mid i \in G_1\} < \min\{\tilde{s}^+_i \mid i \in G_1\} + 3\mu &< \max\{\{\tilde{s}^+_i \mid i \in G_2\} \cup \{\mu\}\} + 3\mu \\ &< \max\{\{\tilde{s}'_i \mid i \in G_2\} \cup \{\mu\}\} + 6\mu.\end{aligned}$$

□

We are now ready for the key lemma in the proof of the main result—the multiplicative decrease of the potential function at the end of each round.

Lemma A.1. *If $x < 1 + \frac{1}{R_2 m^3}$ at the end of round t in Alg-WGS with $R_2 = 288e^2$, then $\Phi(\mathbf{p}_t) \le \Phi(\mathbf{p}_{t-1})(1 - \frac{1}{18e^2m^3})$.*

Proof. We use the following notation. Let $\mathbf{s} = \mathcal{O}(\mathbf{p}_{t-1})$, $\tilde{\mathbf{s}} = \widetilde{\mathcal{O}}(\mathbf{p}_{t-1}, \mu)$, and $\mathbf{s}' = \mathcal{O}(\mathbf{p}_t)$, $\tilde{\mathbf{s}}' = \widetilde{\mathcal{O}}(\mathbf{p}_t, \mu)$. The intuition of the proof is similar to the version with exact precision. By the conditions used to define G_1 and G_2, we always have $\tilde{s}_{i_k} \ge \tilde{s}_{i_1}/e$ and $\tilde{s}_{i_k} - \tilde{s}_{i_{k+1}} > \tilde{s}_{i_k}/(m+1) \ge \tilde{s}_{i_1}/e(m+1)$. Hence, roughly speaking, every good in G_1 has reasonably large surplus, and there is a reasonably large gap between the surpluses in G_1 and G_2. Next, at the end of the current round, we decreased the minimum surplus of a good in G_1 to either $\min\{\tilde{s}'_i \mid i \in G_1\} \approx \mu$ (Case (1) shown later) or $\min\{\tilde{s}'_i \mid i \in G_1\} \approx \max\{\tilde{s}'_i \mid i \in G_2\}$ (Case (2) shown later). In both cases, the total value of Φ must decrease by a factor of $1 - \Omega(1/m^3)$.

More formally, if the algorithm proceeds to round t, then $\|\tilde{\mathbf{s}}\| > \varepsilon'^2$. By the definition of set G_1, we have $\tilde{s}_{i_1} \le (1 + \frac{1}{m})\tilde{s}_{i_2} \le \cdots \le (1 + \frac{1}{m})^{k-1}\tilde{s}_{i_k} < e \cdot \tilde{s}_{i_k}$. Hence, $\tilde{s}^2_{i_k} > (\tilde{s}_{i_1}/e)^2 \ge \Phi(\mathbf{p}_{t-1})/(me^2) >$

$(\varepsilon'/(\sqrt{m}e))^2$, so the surpluses of goods in G_1 are similar up to a factor of e and bounded from below. In addition, we have $(\tilde{s}_{i_k} - \tilde{s}_{i_{k+1}})^2 > (\tilde{s}_{i_1}/e(m+1))^2 \geq \Phi(\mathbf{p}_{t-1})/(e^2(m+1)^2 m)$.

Since we rely on an approximate demand oracle, the surpluses of goods in G_1 might not change in a monotone fashion when increasing their prices. Nevertheless, we can relate the surplus in the beginning and the end of a round as follows. For every $i \in G_1$, by Claim 2.4, the surplus from exact demands satisfies $s'_i \leq x \cdot s_i$. Thus,

$$\tilde{s}'_i \;\leq\; s'_i + \mu \;\leq\; x s_i + \mu \;\leq\; x(\tilde{s}_i + \mu) + \mu \;=\; x\tilde{s}_i + (1+x)\mu.$$

Since $x < 1 + \frac{1}{R_2 m^3}$, it holds that $(1+x)\mu < 3\mu \leq \tilde{s}_i/(R_2 m^3)$. This means that the increase within a round is bounded by $\tilde{s}'_i < (1 + \frac{2}{R_2 m^3})\tilde{s}_i$. Since we do not touch the price of any good $j \in G_2$, the WGS property implies for exact demands $s'_j \geq s_j$. Hence, $\tilde{s}'_j \geq \tilde{s}_j - 2\mu$.

Now, to bound the change of $\Phi(\mathbf{p}_t)$, we consider $\tilde{\mathbf{s}}'$ according to G_1 and G_2. We distinguish two cases:

Case 1: $\max\{\tilde{s}'_i \mid i \in G_2\} < \mu$. Intuitively, in this case, the algorithm has decreased the surplus of some good in G_1 to approximately 0 (recall that μ is sufficiently small). This decrease alone brings down the potential function Φ by a factor of $1 - \Omega(1/m)$. All other surpluses will cause an increase by a factor of at most $1 + O(1/m^3)$.

More formally, Claim A.4 gives us $\mu < \min\{\tilde{s}'_i \mid i \in G_1\} < 7\mu$. Hence, the contribution of goods of G_1 to $\Phi(\mathbf{p}_t)$ can be upper bounded by

$$\sum_{j=1}^{k} \tilde{s}'^2_{i_j} < \sum_{j=1}^{k-1} \left(1 + \frac{2}{R_2 m^3}\right)^2 \tilde{s}^2_{i_j} + 49\mu^2. \tag{10}$$

Furthermore, for every $i \in G_2$, if $-m^3\mu \leq \tilde{s}'_i \leq \mu$, we have $\tilde{s}'^2_i \leq m^6\mu^2$, and if $\tilde{s}'_i < -\mu$, by the WGS property of the market, we know $s_i \leq s'_i \leq \tilde{s}'_i + \mu < 0$. Thus, since $\tilde{s}'_j \geq \tilde{s}_j - 2\mu$,

$$\sum_{\substack{j \in G_2 \\ \tilde{s}'_j < -m^3\mu}} \tilde{s}'^2_j \leq \sum_{\substack{j \in G_2 \\ \tilde{s}'_j < -m^3\mu}} (\tilde{s}_j - 2\mu)^2 \leq \sum_{\substack{j \in G_2 \\ \tilde{s}'_j < -m^3\mu}} \left(1 + \frac{2}{R_2 m^3}\right)^2 \tilde{s}^2_j.$$

Hence, the contribution of goods of G_2 to $\Phi(\mathbf{p}_t)$ can be upper bounded by

$$\sum_{j=k+1}^{m} \tilde{s}'^2_{i_j} \leq \sum_{j=k+1}^{m} \max\left\{\left(1 + \frac{2}{R_2 m^3}\right)^2 \tilde{s}^2_{i_j}, m^6\mu^2\right\} < \sum_{j=k+1}^{m} \left(1 + \frac{2}{R_2 m^3}\right)^2 \tilde{s}^2_{i_j} + m^6\mu^2. \tag{11}$$

Combining Equations (10) and (11),

$$\begin{aligned}
\Phi(\mathbf{p}_t) = \sum_{j=1}^{m} \tilde{s}'^2_{i_j} &< \sum_{j \neq k} \left(1 + \frac{2}{R_2 m^3}\right)^2 \tilde{s}^2_{i_j} + (m^6 + 49)\mu^2 \\
&= \left(1 + \frac{2}{R_2 m^3}\right)^2 (\Phi(\mathbf{p}_{t-1}) - \tilde{s}^2_{i_k}) + (m^6 + 49)\mu^2 \\
&< \left(1 + \frac{2}{R_2 m^3}\right)^2 \left(1 - \frac{1}{e^2 m}\right) \Phi(\mathbf{p}_{t-1}) + \left(\frac{4}{R_1^2 m^7} + \frac{196}{R_1^2 m^{13}}\right) \varepsilon'^2 \\
&< \left(1 - \frac{1}{2e^2 m}\right) \Phi(\mathbf{p}_{t-1}),
\end{aligned}$$

where the last inequality holds for any $m \geq 2$ with our choice of values of R_1, R_2.

Case 2: $\max\{\tilde{s}'_i \mid i \in G_2\} \geq \mu$. Intuitively, in this case, the gap between surpluses in G_1 and G_2 decreases to approximately 0. In the following, we show that the closing this gap yields a decrease of the potential function Φ by a factor of $1 - \Omega(1/m^3)$. All other surpluses will increase by a factor of at most $1 + O(1/m^3)$. In combination, it turns out that Φ will decrease by a factor of $1 - \Omega(1/m^3)$.

More formally, in this case, $\min\{\tilde{s}'_i \mid i \in G_1\} \geq \max\{\tilde{s}'_j \mid j \in G_2\}$. Let $s_{G_1} = \min\{\tilde{s}'_i \mid i \in G_1\}$ and $s_{G_2} = \max\{\tilde{s}'_j \mid j \in G_2\}$. For every $i \in G_1$, let $\tilde{s}'_i = x'\tilde{s}_i - \delta_i$, where $x' = (1 + \frac{2}{R_2 m^3})$, and for every $j \in G_2$, let $\tilde{s}'_j = \tilde{s}_j - 2\mu + \delta_j$. Hence, $\delta_i, \delta_j \geq 0$ for all i, j. Further, we have $|\sum_{i=1}^m \tilde{s}_i| \leq m\mu$ and $|\sum_{i=1}^m \tilde{s}'_i| \leq m\mu$, and hence

$$\begin{aligned}
\sum_{i \in G_1} \delta_i &= \sum_{i \in G_1} x'\tilde{s}_i - \sum_{i \in G_1} \tilde{s}'_i \\
&\geq \sum_{i \in G_1} \tilde{s}_i + \sum_{j \in G_2} \tilde{s}'_j - m\mu = \sum_{i \in G_1} \tilde{s}_i + \sum_{j \in G_2} (\tilde{s}_j + \delta_j - 2\mu) - m\mu \\
&\geq \sum_{j \in G_2} \delta_j - 4m\mu
\end{aligned}$$

and

$$\sum_{i \in G_1} \delta_i \geq \frac{1}{2}(\tilde{s}_{i_k} - \tilde{s}_{i_{k+1}} - 4m\mu) \geq \frac{1}{4}(\tilde{s}_{i_k} - \tilde{s}_{i_{k+1}}).$$

Now we have

$$\begin{aligned}
\Phi(\mathbf{p}_t) &= \sum_i \tilde{s}'^2_i = \sum_{i \in G_1} (x'\tilde{s}_i - \delta_i)^2 + \sum_{j \in G_2} (\tilde{s}_j - 2\mu + \delta_j)^2 \\
&= \left(\sum_{i \in G_1} x'^2 \tilde{s}_i^2 + \sum_{j \in G_2} (\tilde{s}_j - 2\mu)^2 \right) + \left(\sum_{j \in G_2} \delta_j (\tilde{s}_j - 2\mu + \delta_j) - \sum_{i \in G_1} \delta_i (x'\tilde{s}_i - \delta_i) \right) - \sum_{i \in G_1} x'\tilde{s}_i \delta_i \\
&\quad + \sum_{j \in G_2} \delta_j (\tilde{s}_j - 2\mu) \qquad (12) \\
&< \left(x'^2 \Phi(\mathbf{p}_{t-1}) - 4\mu \sum_{j \in G_2} \tilde{s}_j + 4m\mu^2 \right) + \left(s_{G_2} \sum_{j \in G_2} \delta_j - s_{G_1} \sum_{i \in G_1} \delta_i \right) - \tilde{s}_{i_k} \sum_{i \in G_1} \delta_i \\
&\quad + (\tilde{s}_{i_{k+1}} - 2\mu) \sum_{j \in G_2} \delta_j \qquad (13) \\
&< x'^2 \Phi(\mathbf{p}_{t-1}) - 4\tilde{s}_m m\mu + 4m\mu^2 + 4s_{G_1} m\mu + 4\tilde{s}_{i_{k+1}} m\mu - (\tilde{s}_{i_k} - \tilde{s}_{i_{k+1}}) \sum_{i \in G_1} \delta_i \qquad (14) \\
&< x'^2 \Phi(\mathbf{p}_{t-1}) + 4m\mu^2 + 24m^2\mu - \frac{1}{4}(\tilde{s}_{i_k} - \tilde{s}_{i_{k+1}})^2 \qquad (15) \\
&< \left(1 + \frac{4}{R_2 m^3} + \frac{4}{R_2^2 m^6} + \frac{16}{R_1^2 m^{12}} + \frac{96}{R_1 m^4} - \frac{1}{4e^2 (m+1)^2 m} \right) \Phi(\mathbf{p}_{t-1}) \qquad (16) \\
&= \left(1 - \frac{1}{18e^2 m^3} \right) \Phi(\mathbf{p}_{t-1}). \qquad (17)
\end{aligned}$$

Here, (12) can be derived by expanding the quadratic formula and appropriately reorganizing the terms. For the step from (12) to (13), in the first bracket we overestimate the quadratic terms of $\tilde{s}$ into $x'^2\Phi(\mathbf{p}_{t-1})$ and $|G_2|$ by m. In the second bracket, we return to $\tilde{s}'_i$ and $\tilde{s}'_j$, which in turn are bounded correctly using s_{G_1} for all $i \in G_1$ and s_{G_2} for all $j \in G_2$. For the final two terms in (12) and (13), we use the definition of $\tilde{s}_{i_k}$ and $\tilde{s}_{i_{k+1}}$ and the fact that $x' > 1$. For the step from (13) to (14), for the first bracket of (13) we use $\tilde{s}_j \geq \tilde{s}_m$ for every $j \in G_2$. For the second bracket of (13), we note $\tilde{s}_{G_1} \geq \tilde{s}_{G_2}$ and the difference between the sums of δ-terms is bounded by $4m\mu$ as noted earlier. By the same argument, we can transform the last two terms of (13) as shown. Note that we simply drop $-2\mu \sum_{j \in G_2} \delta_j < 0$. From (14) to (15), we use the fact that every surplus is bounded by $2m$ in its absolute value by Claim A.1. For the last term, we use the bound for $\sum_{i \in G_1} \delta_i$ as noted earlier. From (15) to (16), we just replace μ by its definition and use the bound $\Phi(\mathbf{p}_{t-1}) > \varepsilon'^2$ and $(\tilde{s}_{i_k} - \tilde{s}_{i_{k+1}})^2 > \Phi(\mathbf{p}_{t-1})/(e^2(m+1)^2 m)$ as shown earlier.

Finally, the multiplicative term in (16) can be decreased to strictly less than 1 for every $m \geq 2$ with our choice of values of R_1 and R_2. The final expression in (17) proves the lemma. □

The previous lemma shows a decrease in the potential only for rounds in which the x determined by binary search is rather small. Lemma 2.2 continues to hold here and bounds the number of rounds with a larger value of x. The following variant differs only in the constant R_2, and its proof is literally the same as for Lemma 2.2.

Lemma A.2. *During a run of Alg-WGS, there can be only $O(m^4 D_1)$ many rounds that end with $x \geq 1 + \frac{1}{R_2 m^3}$.*

Finally, we can assemble the properties to show the main result.

Theorem A.1. *For any market that satisfies Assumptions 2.1 and 2.2, and for any $\varepsilon > 0$, Alg-WGS returns the price vector of an $(1+\varepsilon)$-approximate market equilibrium in time polynomial in the input size and* $\log 1/\varepsilon$.

Proof. Let x_t be the value of x we find in round t of Alg-WGS. First, because at least one price will increase by a factor of x_t in round t, by Claim A.3 we have $\prod_t x_t \leq 2^{mD_1}$. At the end of round t, if $x_t \geq 1 + \frac{1}{R_2 m^3}$, let $s = \max\{s_i \mid s_i \in \widetilde{O}(\mathbf{p}_{t-1})\}$ and $s' = \max\{s_i \mid s_i \in \widetilde{O}(\mathbf{p}_t)\}$, then by Claim 2.4 we have $\Phi(\mathbf{p}_t) \leq ms'^2 \leq mx_t^2 s^2 \leq mx_t^2 \Phi(\mathbf{p}_{t-1})$. Moreover, by Lemma A.2, there will be at most $O(m^4 D_1)$ such rounds. Hence, the total increase of $\Phi(\mathbf{p}_t)$ in these rounds will be no more than a factor of $\prod_{x_t \geq 1+1/R_2 m^3} mx_t^2 \leq m^{O(m^4 D_1)} 2^{2mD_1} = m^{O(m^4 D_1)}$.

For all other rounds, we have $x < 1 + \frac{1}{R_2 m^3}$, and by Lemma A.1, the potential function is decreased by a factor of $1/(1 - \Omega(\frac{1}{m^3}))$. Therefore, the total number of rounds before $\Phi(\mathbf{p}_t) \leq \varepsilon'^2$ will be at most

$$O\left(\log_{1/(1-\Omega(\frac{1}{m^3}))} \frac{m^{O(m^4 D_1)}}{\varepsilon'^2}\right) = O\left(D_1 m^7 \log m + m^3 \log \frac{1}{\varepsilon}\right).$$

In each round, the number of queries to the oracle is no more than $O(\log \Delta) = O(D_1 + D_2 + \log m + \log \frac{1}{\varepsilon})$. We conclude that the total number of queries during the algorithm is $O((D_1 m^7 \log m + m^3 \log \frac{1}{\varepsilon})\ (D_1 + D_2 + \log m + \log \frac{1}{\varepsilon}))$, which is a polynomial in the input size and $\log 1/\varepsilon$. □

B EXACT EQUILIBRIUM FOR EXCHANGE MARKETS WITH SPENDING CONSTRAINT UTILITIES

Here we show how to obtain an exact market equilibrium in exchange spending constraint markets. Using the Alg-Spending-Exact framework, we convert the approximate market equilibrium obtained in Section 3.2 into an exact equilibrium. To achieve this, we rely on the full information of the spending constraint utilities. This step is an extension of the technique developed in the work of Duan and Mehlhorn [26] for the linear exchange markets. However, there are several challenges due to the much more involved setting of spending constraint utilities, where the allocated partitions make the remaining budgets of agents and the values of goods dependent on too many parameters. In the following, we present how to handle the extra complexity of the problem, and this result resolves an open question of Duan and Mehlhorn [26] of finding an exact polynomial-time algorithm for exchange markets with spending constraint utilities.

Let $\mathbf{p}$ be the price vector of an $(1+\varepsilon)$-approximate equilibrium. We first construct a bipartite graph $EG'(\mathbf{p}) = (A \cup G, E)$, where the edge set E is a union of equality edges in $EG(\mathbf{p})$, edges due to positive endowments, and edges due to allocated segments. The main idea here is to construct a set of components of agents and goods such that there is no interaction across the components. In addition, we want at least one good with price 1 in every such component. To achieve this latter condition, whenever there is a component C of $EG'(\mathbf{p})$ without a good with price 1, we raise the prices of goods in C by a common factor $x > 1$ until a new equality edge appears. By Assumption 3.1, a new equality edge will always emerge in during this procedure because prices of goods in C are increasing, which makes goods outside C more and more attractive to the agents in C.

The next lemma shows that the updated price vector after the price increase still remains a $(1+\varepsilon)$-approximate equilibrium.

Lemma B.1. *The price vector* $\mathbf{p}$ *at the end of the while loop in* Alg-Spending-Exact *is a* $(1+\varepsilon)$-*approximate equilibrium.*

Proof. Note that we increase prices of goods in a component C when each good has price greater than 1. Corollary 3.2 implies that the surplus of each good in C is 0. Hence, the old allocation will still be feasible after the price change, and the surpluses remain the same. Therefore, the updated $\mathbf{p}$ will remain a $(1+\varepsilon)$-approximate equilibrium. □

At this stage, we can assume that each component of $EG'(\mathbf{p})$ has a good with price 1. We then work on each component of $EG'(\mathbf{p})$ separately. We assume for convenience that $EG'(\mathbf{p})$ is a single component.

Next we set up a system of linear equations in price variables of the form $A\mathbf{p} = b$ and show that the matrix A has full rank. Finally, we show that by perturbing the vector b slightly, we can get an exact equilibrium. Consider the components of $EG(\mathbf{p})$, for instance, after removing edges due to endowment and allocated segments from $EG'(\mathbf{p})$. Let $C_1, \ldots, C_K$ be the set of components of $EG(\mathbf{p})$. In each C_l, $1 \le l \le K$, all goods are connected with each other through a set of equality edges. Whenever there are two current segments (i, j, k) and (i, j', k') of the same agent i, we have the following relation between the prices of goods j and j':

$$u_{ijk} p_{j'} = u_{ij'k'} p_j. \tag{18}$$

This implies that for each component C_l, $|C_l \cap G| - 1$ of these equations are linearly independent, and there is essentially one free price variable. Further, since there is no money flow across components with respect to the current allocations, we have the budget balance condition for each component:

Remaining worth of goods − remaining budgets of agents (after allocated segments)
= sum of surpluses.

For component C_l, the condition reads

$$\sum_{j \in C_l \cap G} \left(p_j - \sum_{(i,j,k) \in F} B_{ijk} \sum_{j'} w_{ij'} p_{j'} \right) - \sum_{i \in C_l \cap A} \left(\sum_{j'} w_{ij'} p_{j'} - \sum_{(i,j,k) \in F} B_{ijk} \sum_{j'} w_{ij'} p_{j'} \right) = \sum_{j \in C_l \cap G} \varepsilon_j,$$

ALGORITHM 5: ALG-SPENDING-EXACT

Input: Exchange market with a set A of agents and a set G of goods; w_{ij}, u_{ijk}, B_{ijk} are market parameters as defined in Section 1.1

Output: Prices $\mathbf{p}$ of an exact market equilibrium

$m \leftarrow |G|$; $n \leftarrow |A|$; $L \leftarrow$ total bit length of all input parameters; $\varepsilon \leftarrow 1/m^{4m}2^{4m^2L}$

$\mathbf{p} \leftarrow (1+\varepsilon)$-approximate equilibrium using ALG-SPENDINGwith ALG-SPENDING-ROUNDING

$\mathbf{s} \leftarrow O(\mathbf{p})$. If $\mathbf{s} = (0, 0, \ldots, 0)$, then **return p**.

$EG(\mathbf{p}) \leftarrow$ (undirected) equality graph at prices $\mathbf{p}$ /*as defined at the beginning of Section 3*/

$F \leftarrow \{(i,j,k) \mid (i,j,k)$ is an allocated segment$\}$

$EG'(\mathbf{p}) \leftarrow EG(\mathbf{p}) \cup \{(i,j) \mid w_{ij} > 0\} \cup \{(i,j) \mid (i,j,k) \in F$ for some $k\}$.

while $EG'(\mathbf{p})$ *contains a connected component* C *that does not has a good with price* 1 **do**

  Binary search the smallest $x > 1$ such that $EG(\mathbf{p}) \subset EG(\text{UPDATE}(\mathbf{p}, x, C))$.

  $\mathbf{p} \leftarrow \text{UPDATE}(\mathbf{p}, x, C)$.

  Recompute $EG(\mathbf{p})$ and $EG'(\mathbf{p})$.

end

/* Without loss of generality, we assume that $EG'(\mathbf{p})$ consists of only one connected component. If there are more than one, then apply the following procedure individually to each component */

Let $C_1, \ldots, C_K$ be the connected components of $EG(\mathbf{p})$

Set up the following system of linear equations in price variable

(1) $p'_i = 1$ for a good i whose price is 1

(2) For each component $C_l, 1 \leq l \leq K$

  a. $|C_l| - 1$ linearly independent equations of the form $u_{ijk}p'_{j'} = u_{ij'k'}p'_j$, where (i,j,k) and (i,j',k') are the current segments.

  b. $\sum_{j \in C_l \cap G} p'_j - \sum_j R_{lj}p'_j = 0$, where
  $R_{lj} = \sum_{i \in C_l \cap A} w_{ij}(1 - \sum_{(i,j',k) \in F; j' \notin C_l \cap G} B_{ij'k}) + \sum_{i \notin C_l \cap A} w_{ij} \sum_{(i,j',k) \in F; j' \in C_l \cap G} B_{ij'k}$

$\mathbf{p}' \leftarrow$ the solution of the preceding system of equations

return p′

where F is the set of allocated segments. Rearranging the preceding equation, we get

$$\sum_{j \in C_l \cap G} p_j - \sum_{j \in G} p_j R_{lj} = \sum_{j \in C_l \cap G} \varepsilon_j, \tag{19}$$

$$\text{where } R_{lj} = \sum_{i \in C_l \cap A} w_{ij} \left(1 - \sum_{(i,j',k) \in F; j' \notin C_l \cap G} B_{ij'k}\right) + \sum_{i \notin C_l \cap A} w_{ij} \sum_{(i,j',k) \in F; j' \in C_l \cap G} B_{ij'k},$$

Each R_{lj} is a rational number with denominator at most 2^{2L}, where L is the total bit length of all input parameters w_{ij}, u_{ijk} and B_{ijk}.

LEMMA B.2. *For every* $1 \leq l \leq K$ *and* $j \in C_l \cap G$, $0 \leq R_{lj} \leq 1$. *For every* $j \in G$, $\sum_l R_{lj} = 1$.

PROOF. We have $\sum_i w_{ij} = 1, \forall j$. Further, both $\sum_{(i,j',k) \in F; j' \notin C_l \cap G} B_{ij'k}$ and $\sum_{(i,j',k) \in F; j' \in C_l} B_{ij'k}$ take values in $[0, 1]$, and hence the first claim of the lemma follows. For the second claim, $\sum_l R_{lj} = \sum_i w_{ij} = 1$. □

Let M be the coefficient matrix of the system of equations (19). Then

$$M_{lj} = \begin{cases} 1 - R_{lj} & \text{if } j \in C_l \cap G \\ -R_{lj}, & \text{otherwise.} \end{cases}$$

From Lemma B.2, each column j of M has at most one positive entry, namely M_{lj}, where $j \in C_l \cap G$, and each column of M sums to 0. There are in total L equations of type (19), one for each component. Next we eliminate the equations of type (18). Then there will be only one price variable per component. We designate a representative good for each component, say good l for C_l. Then each price p_j in C_l is a constant multiple of price p_l of good l. Let $p_j = \alpha_j p_l$, where α_j is a rational number whose numerator and denominator are products of at most m u_{ijk}'s. Now we can rewrite the budget balance equation (19) for C_l in terms of L price variables as follows:

$$p_l \sum_{j \in C_l \cap G} \alpha_j - \sum_{l'} p_{l'} \sum_{j \in C_{l'} \cap G} \alpha_j R_{lj} = \sum_{j \in C_l \cap G} \varepsilon_j.$$

Let $T_l = \sum_{j \in C_l \cap G} \alpha_j$, $S_{ll'} = \sum_{j \in C_{l'} \cap G} \alpha_j R_{lj}$, and $\varepsilon_l = \sum_{j \in C_l \cap G} \varepsilon_j$. The preceding equation becomes

$$p_l T_l - \sum_{l'} p_{l'} S_{ll'} = \varepsilon_l. \tag{20}$$

Let N be the coefficient matrix of the system of equations (20). Then

$$N_{ll'} = \begin{cases} T_l - S_{ll} & \text{if } l = l' \\ -S_{ll'}, & \text{otherwise.} \end{cases}$$

Since both T_l and $S_{ll'}$ are rational numbers with denominator at most 2^{mL} and 2^{2mL}, respectively, each $N_{ll'}$ is a rational number with denominator at most 2^{3mL}.

Lemma B.3. *$0 \le N_{ll}$ for every l, $N_{ll'} \le 0$ for every $l \neq l'$, and $\sum_l N_{ll'} = 0$ for every l'.*

Proof. The proof essentially follows using Lemma B.2. The first two claims are straightforward, and for the last claim we have

$$\sum_l S_{ll'} = \sum_l \sum_{j \in C_{l'} \cap G} \alpha_j R_{lj} = \sum_{j \in C_{l'} \cap G} \alpha_j \sum_l R_{lj} = \sum_{j \in C_{l'} \cap G} \alpha_j = T_{l'}. \qquad \square$$

Since there is a good i with price 1, we assume without loss of generality that good i belongs to component K, and hence $\alpha_K p_K = 1$. The next lemma is an adaptation of a result of Duan and Mehlhorn [26].

Lemma B.4. *The K equations consisting of Equation (20) for components $1, \ldots, K-1$ and the equation $\alpha_K p_K = 1$ are linearly independent.*

Proof. Let N' be the coefficient matrix of this system of equations. It is easy to check that N' is the same as N except $N'_{Li} = 0$, $1 \le i \le K-1$. Assume by contradiction that there is a non-zero vector $a = (a_1, \ldots, a_K)$ such that $a^T N' = 0$. Let a_{l_0} be the entry in $\{a_1, \ldots, a_{K-1}\}$ that has the largest absolute value, and without loss of generality we assume that $a_{l_0} > 0$ and the first K' entries of a are equal to a_{l_0}, i.e., $a_1 = \cdots = a_{K'} = a_{l_0}$.

For each $l \le K'$, we have

$$\begin{aligned} 0 &= \sum_{1 \le h < K} a_h N_{hl} + a_K \cdot 0 \\ &= a_{l_0} \sum_{1 \le h \le K} N_{hl} - a_{l_0} N_{Kl} + \sum_{K' < h < K} (a_h - a_{l_0}) N_{hl} \\ &= -a_{l_0} N_{Kl} + \sum_{K' < h < K} (a_h - a_{l_0}) N_{hl}. \end{aligned}$$

Using Lemma B.3, the preceding implies that $N_{hl} = 0$ for $K' < h \le K$ and $l \le K'$. Next we show that $N_{lh} = 0$ for $1 \le l \le K'$ and $K' < h \le K$ as well. By summing up the Equations (20) for $1 \le l \le K'$, we get

$$\begin{aligned}
\sum_{l\le K'} \varepsilon_l &= \sum_{l\le K'} N_{ll}p_l + \sum_h \sum_{l\le K';l\ne h} N_{lh}p_h \\
&= \sum_{l\le K'} N_{ll}p_l + \sum_{h\le K'} \sum_{l\le K';l\ne h} N_{lh}p_h + \sum_{h>K'} \sum_{l\le K'} N_{lh}p_h \\
&= \sum_{h\le K'} N_{hh}p_h + \sum_{h\le K'} \sum_{l\ne h} N_{lh}p_h + \sum_{h>K'} \sum_{l\le K'} N_{lh}p_h \\
&= \sum_{h\le K'} p_h \sum_l N_{lh} + \sum_{h>K'} \sum_{l\le K'} N_{lh}p_h \\
&= \sum_{h>K'} \sum_{l\le K'} N_{lh}p_h.
\end{aligned}$$

Since $p_h \ge 1$ for every h, if some N_{lh} is non-zero for $l \le K'$ and $h > K'$, then the right-hand side is at most $-1/2^{3mL}$, which is a contradiction. It implies that $N_{lh} \ne 0$ if and only if both l and h are either less than or equal to K' or larger than K'. Let A_1 and G_1 denote the set of agents and goods of components $C_1, \ldots, C_{K'}$, respectively. Let $A_2 = A \setminus A_1$ and $G_2 = G \setminus G_1$. We can further conclude the following:

- $w_{ij} = 0$ for every $i \in A_1,\ j \in G_2$, and otherwise $N_{lh} \ne 0$ for $l \le K'$ and $h > K'$. Similarly, $w_{ij} = 0$ for every $i \in A_2,\ j \in G_1$.
- Agents in A_1 have no allocated goods in G_2, and otherwise budget balance equations of components $C_l, l > K'$ will have some non-zero $p_j, 1 \le j \le K'$ and that will make $N_{lh} \ne 0$ for $l > K'$ and $h \le K'$. Similarly, agents in A_2 have no allocated goods in G_1.

This is impossible because we have assumed that $EG'(\mathbf{p})$ consists of one single component. Therefore, N' must have full rank. □

Overall, we have established that Equations of (18), (19), and $p_i = 1$ are linearly independent. We can write this system in the matrix form as $A\mathbf{p} = b$, where A is invertible and all entries are rational numbers with common denominator at most 2^{2L}.

Consider the system $A\mathbf{p}' = b'$ for a price vector $\mathbf{p}'$, where b' is a unit vector with a one in the row corresponding to the equation $p_i = 1$. Next we show that $\mathbf{p}'$ gives an exact equilibrium. For that, we need to show the following:

(a) Equality edges with respect to $\mathbf{p}'$ and $\mathbf{p}$ are the same. This will imply that all allocated segments remain allocated.

(b) $(s, A \cup G \cup t)$ is a min-cut in $N(\mathbf{p}')$. Combining the equation (2b) in step 13 of Alg-Spending-Exact, this will imply that there is a feasible allocation on current segments that gives surplus of each good.

By Cramer's rule and Lemma B.2, the solution of $A\mathbf{p}' = b'$ is a vector of rational numbers with common denominator $D \le m^m 2^{2m(m+1)L}$. In other words, all p'_i are of form q_i/D, where q_i, D are integers. Since $||b| - |b'|| \le 2\varepsilon$, we have $|p'_i - p_i| \le 2\varepsilon D$ for every i. Let $\varepsilon' = 2D^2\varepsilon$, then $|Dp_i - q_i| = D|p_i - p'_i| \le 2\varepsilon D^2 = \varepsilon'$. For part (a), suppose $u_{ijk}p_{j'} \le u_{ij'k'}p_j$ and then

$$\begin{aligned}
u_{ijk}q_{j'} \le u_{ijk}(Dp_{j'} + \varepsilon') \le Du_{ij'k'}p_j + u_{ijk}\varepsilon' &\le u_{ij'k'}q_j + (u_{ij'k'} + u_{ijk})\varepsilon' \\
&< u_{ij'k'}q_j + 1.
\end{aligned}$$

Since both $u_{ijk}q_{j'}$ and $u_{ij'k'}q_j$ are integers, we have $u_{ijk}p'_{j'} \leq u_{ij'k'}p'_j$. This implies that all equality edges with respect to $\mathbf{p}'$ and $\mathbf{p}$ are the same. That further implies that all allocated segments remain allocated.

For part (b), consider the network $N(\mathbf{p}')$ with respect to prices $\mathbf{p}'$. Recall that in $N(\mathbf{p}')$,

- the capacity of edge from source node s to agent i is $\sum_{j'} w_{ij'}p'_{j'}(1 - \sum_{j,k:(i,j,k)\in F} B_{ijk})$, and
- the capacity of a MBB edge from agent i to good j is $B_{ijk} \sum_{j'} w_{ij'}p'_{j'}$.

Since p'_j's are all rational numbers with a common denominator $D \leq m^m 2^{2m(m+1)L}$, all capacities in $N(\mathbf{p}')$ are rational numbers with a common denominator no more than D^2. In addition, because $|p'_i - p_i| \leq 2\varepsilon D$, the capacity of each edge e in $N(\mathbf{p})$ is at most the capacity of e in $N(\mathbf{p}')$ plus $4\varepsilon D$. Let c be the capacity of cut $(s, A \cup G \cup t)$ in $N(\mathbf{p}')$. Suppose there is a min cut in $N(\mathbf{p}')$ with value less than c. Then that value is at most $c - 1/D^2$. This same cut in $N(\mathbf{p})$ will have value at most $c - 1/D^2 + (m + n + mn)4D\varepsilon$. In addition, the capacity of the cut $(s, A \cup G \cup t)$ in $N(\mathbf{p})$ is at least $c - 4n\varepsilon D$. Therefore, the total surplus of goods in $N(\mathbf{p})$ is at least

$$c - 4n\varepsilon D - \left(c - \frac{1}{D^2}\right) - (m + n + mn)4\varepsilon D = \frac{1}{D^2} - 4(m + 2n + mn)\varepsilon D > \varepsilon,$$

which is a contradiction. Hence, condition (b) also holds.

We conclude with our main theorem.

THEOREM B.5. *For every spending constraint exchange market satisfying Assumption 3.1, ALG-SPENDING-EXACT returns the price vector of a market equilibrium in time polynomial in m and L.*

PROOF. By Lemma 3.9, we know that we arrive at a $(1 + \varepsilon)$-approximate equilibrium in time polynomial in m, L, and $\log 1/\varepsilon$. For the exact equilibrium, ALG-SPENDING-EXACT solves a system of $K \leq m$ linear equations whose entries are polynomially bounded in m and L, and hence can be done in polynomial time. Further, $\log 1/\varepsilon$ is a polynomial of m and L as $\varepsilon = 1/m^{4m}2^{4m^2 L}$. Since M is a polynomial of m, L, binary search and ROUNDING run in polynomial time in each round of the framework. Hence, the total running time of the algorithm is polynomial in m and L. ◻

REFERENCES

[1] Nima Anari, Tung Mai, Shayan Oveis Gharan, and Vijay Vazirani. 2018. Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. In *Proceedings of the 29th Symposium on Discrete Algorithms (SODA'18)*. 2274–2290.

[2] Kenneth Arrow and Gerard Debreu. 1954. Existence of an equilibrium for a competitive economy. *Econometrica* 22, 3 (1954), 265–290.

[3] Kenneth Arrow and Leonid Hurwicz. 1958. On the stability of the competitive equilibrium. *Econometrica* 26, 4 (1958), 522–552.

[4] Xiaohui Bei, Jugal Garg, and Martin Hoefer. 2016. Ascending-price algorithms for unknown markets. In *Proceedings of the 17th Conference on Economics and Computation (EC'16)*. 699.

[5] Xiaohui Bei, Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. 2017. Earning limits in Fisher markets with spending-constraint utilities. In *Proceedings of the 10th Symposium on Algorithmic Game Theory (SAGT'17)*. 67–79.

[6] 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'11)*. 127–136.

[7] Michael Blad. 1978. On the speed of adjustment in the classical tatonnement process: A limit result. *Journal of Economic Theory* 19, 1 (1978), 186–191.

[8] William Brainard and Herbert Scarf. 2000. *How to Compute Equilibrium Prices in 1891.* Discussion Paper 1270. Cowles Foundation.

[9] 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 on Foundations of Computer Science (FOCS'09)*. 273–282.

[10] Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. 2017. The complexity of non-monotone markets. *Journal of the ACM* 64, 3 (2017), Article 20, 56 pages.

[11] Yun Kuen Cheung and Richard Cole. 2016. A unified approach to analyzing asynchronous coordinate descent and tatonnement. arXiv:1612.09171.

[12] Yun Kuen Cheung, Richard Cole, and Nikhil Devanur. 2013. Tatonnement beyond gross substitutes? Gradient descent to the rescue. In *Proceedings of the 45th Symposium on Theory of Computing (STOC'13)*. 191–200.

[13] Yun Kuen Cheung, Richard Cole, and Ashish Rastogi. 2012. Tatonnement in ongoing markets of complementary goods. In *Proceedings of the 13th Conference on Electronic Commerce (EC'12)*. 337–354.

[14] Bruno Codenotti, Benton McCune, Sriram Penumatcha, and Kasturi R. Varadarajan. 2005. Market equilibrium for CES exchange economies: Existence, multiplicity, and computation. In *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05)*. 505–516.

[15] Bruno Codenotti, Benton McCune, and Kasturi Varadarajan. 2005. Market equilibrium via the excess demand function. In *Proceedings of the 37th Symposium on Theory of Computing (STOC'05)*. 74–83.

[16] Bruno Codenotti, Sriram Pemmaraju, and Kasturi Varadarajan. 2005. On the polynomial time computation of equilibria for certain exchange economies. In *Proceedings of the 16th Symposium on Discrete Algorithms (SODA'05)*. 72–81.

[17] Bruno Codenotti and Kasturi Varadarajan. 2007. Computation of market equilibria by convex programming. In *Algorithmic Game Theory*, N. Nisan, É. Tardos, T. Roughgarden, and V. Vazirani (Eds.). Cambridge University Press, New York, NY, 135–158.

[18] Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay Vazirani, and Sadra Yazdanbod. 2017. Convex program duality, Fisher markets, and Nash social welfare. In *Proceedings of the 18th Conference on Economics and Computation (EC'17)*. 459–460.

[19] Richard Cole and Lisa Fleischer. 2008. Fast-converging tatonnement algorithms for one-time and ongoing market problems. In *Proceedings of the 40th Symposium on Theory of Computing (STOC'08)*. 315–324.

[20] Richard Cole and Vasilis Gkatzelis. 2015. Approximating the Nash social welfare with indivisible items. In *Proceedings of the 47th Symposium on Theory of Computing (STOC'15)*. 371–380.

[21] Nikhil Devanur, Jugal Garg, and László Végh. 2016. A rational convex program for linear Arrow-Debreu markets. *ACM Transactions on Economics and Computation* 5, 1 (2016), Article 6, 13 pages.

[22] Nikhil Devanur and Ravi Kannan. 2008. Market equilibria in polynomial time for fixed number of goods or agents. In *Proceedings of the 49th Symposium on Foundations of Computer Science (FOCS'08)*. 45–53.

[23] 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), Article 22, 18 pages.

[24] Nikhil Devanur and Vijay Vazirani. 2004. The spending constraint model for market equilibrium: Algorithmic, existence and uniqueness results. In *Proceedings of the 36th Symposium on Theory of Computing (STOC'04)*. 519–528.

[25] Ran Duan, Jugal Garg, and Kurt Mehlhorn. 2016. An improved combinatorial polynomial algorithm for the linear Arrow-Debreu market. In *Proceedings of the 27th Symposium on Discrete Algorithms (SODA'16)*. 90–106.

[26] Ran Duan and Kurt Mehlhorn. 2015. A combinatorial polynomial algorithm for the linear Arrow-Debreu market. *Information and Computation* 243 (2015), 112–132.

[27] B. Curtis Eaves. 1976. A finite algorithm for the linear exchange model. *Journal of Mathematical Economics* 3 (1976), 197–203.

[28] B. Curtis Eaves. 1985. Finite solution of pure trade markets with Cobb-Douglas utilities. In *Economic Equilibrium: Model Formulation and Solution*. Mathematical Programming Studies, Vol. 23. Springer, 226–239.

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

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

[31] Lisa Fleischer, Rahul Garg, Sanjiv Kapoor, Rohit Khandekar, and Amin Saberi. 2016. A simple and efficient algorithm for computing market equilibria. *ACM Transactions on Algorithms* 12, 3 (2016), Article 34, 15 pages.

[32] Jugal Garg, Martin Hoefer, and Kurt Mehlhorn. 2018. Approximating the Nash social welfare with budget-additive valuations. In *Proceedings of the 29th Symposium on Discrete Algorithms (SODA'18)*. 2326–2340.

[33] Jugal Garg, Ruta Mehta, Milind Sohoni, and Vijay Vazirani. 2015. A complementary pivot algorithm for market equilibrium under separable, piecewise-linear concave utilities. *SIAM Journal on Computing* 44, 6 (2015), 1820–1847.

[34] Jugal Garg, Ruta Mehta, Milind A. Sohoni, and Nisheeth K. Vishnoi. 2013. Towards polynomial simplex-like algorithms for market equilibria. In *Proceedings of the 24th Symposium on Discrete Algorithms (SODA'13)*. 1226–1242.

[35] Jugal Garg, Ruta Mehta, and Vijay Vazirani. 2014. Dichotomies in equilibrium computation, and complementary pivot algorithms for a new class of non-separable utility functions. In *Proceedings of the 46th Symposium on Theory of Computing (STOC'14)*. 525–534.

[36] Rahul Garg and Sanjiv Kapoor. 2006. Auction algorithms for market equilibrium. *Mathematics of Operations Research* 31, 4 (2006), 714–729.

[37] Rahul Garg and Sanjiv Kapoor. 2007. Market equilibrium using auctions for a class of gross-substitute utilities. In *Proceedings of the 3rd Workshop on Internet and Network Economics (WINE'07)*. 356–361.

[38] Sungjin Im, Janardhan Kulkarni, and Kamesh Munagala. 2018. Competitive algorithms from competitive equilibria: Non-clairvoyant scheduling under polyhedral constraints. *Journal of the ACM* 65, 1 (2018), Article 3, 33 pages.
[39] Kamal Jain. 2007. A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM Journal on Computing* 37, 1 (2007), 306–318.
[40] Frank Kelly and Vijay Vazirani. 2002. Rate Control as a Market Equilibrium. Retrieved April 10, 209 from http://www.cc.gatech.edu/~vazirani/KV.pdf.
[41] Andreu Mas-Colell, Michael Whinston, and Jerry Green. 1995. *Microeconomic Theory*. Oxford University Press.
[42] Robert Maxfield. 1997. General equilibrium and the theory of directed graphs. *Journal of Mathematical Economics* 27, 1 (1997), 23–51.
[43] James Orlin. 2010. Improved algorithms for computing Fisher's market clearing prices. In *Proceedings of the 42nd Symposium on Theory of Computing (STOC'10)*. 291–300.
[44] Paul Samuelson. 1948. Consumption theory in terms of revealed preference. *Economica* 15, 60 (1948), 243–253.
[45] Herbert Scarf. 1973. *The Computation of Economic Equilibria*. Yale University Press, New Haven, CT.
[46] H. Uzawa. 1960. Walras' tatonnement in the theory of exchange. *Review of Economic Studies* 27, 3 (1960), 182–194.
[47] Hal Varian. 1982. The non-parametric approach to demand analysis. *Econometrica* 50 (1982), 945–974.
[48] Hal Varian. 2005. Revealed preference. In *Samuelsonian Economics and the Twenty-First Century*, M. Szenberg, L. Ramrattan, and A. Gottesman (Eds.). Oxford University Press, Oxford, UK, 99–115.
[49] Vijay Vazirani. 2010. Spending constraint utilities with applications to the Adwords market. *Mathematics of Operations Research* 35, 2 (2010), 458–478.
[50] Vijay Vazirani and Mihalis Yannakakis. 2011. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of the ACM* 58, 3 (2011), 10.
[51] László Végh. 2016. A strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. *SIAM Journal on Computing* 45, 5 (2016), 1729–1761.
[52] Léon Walras. 1874. *Éléments d'économie politique pure, ou théorie de la richesse sociale (Elements of Pure Economics, or the Theory of Social Wealth)*. Lausanne, Paris.
[53] Yinyu Ye. 2007. Exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. *Theoretical Computer Science* 378, 2 (2007), 134–142.

Received April 2018; revised November 2018; accepted March 2019