Market Equilibrium via Indirect Utility Functions: A Tâtonnement Algorithm

Rahul Garg [*] *Sanjiv Kapoor* [†] *Rohit Khandekar* [‡]

Abstract

We give a new mathematical formulation of market equilibria in exchange economies using an *indirect utility function*: the function of prices and income that gives the maximum utility achievable. The formulation is a *convex* program and can be solved when the indirect utility function is convex in prices. We illustrate that many economies including

- Quasi-linear utilities in exchange economies — here there is one good for which every trader has linear utility
- Homogeneous utilities of degree $\alpha \in [0, 1]$ in Fisher economies — this includes Linear, Leontief, Cobb-Douglas, CES, and *resource allocation utilities* like multi-commodity flows

satisfy this condition and can be efficiently solved.

Further, we give a natural tâtonnement price-adjusting algorithm in these economies. Our algorithm, which is applicable to a larger class of utility functions than previously known *weak gross substitutes*, mimics the natural dynamics for the markets as suggested by Walras: it iteratively adjusts a good's price upward when the demand for that good under current prices exceeds its supply; and downward when its supply exceeds its demand. The algorithm computes an approximate equilibrium in a number of iterations that is independent of the number of traders and is almost *linear* in the number of goods.

[*] IBM T.J. Watson Research Center. grahul@us.ibm.com.

[†] Department of Computer Science, Illinois Institute of Technology, kapoor@iit.edu.

[‡] IBM T.J. Watson Research Center. rohitk@us.ibm.com.

1 Introduction

Computation of the Market Equilibrium has recently emerged as an interesting algorithmic problem. The underlying model, common in economics, is that of a market with m traders and n goods, where the traders are endowed with money or/and goods and wish to optimize their utilities. Market equilibrium is defined by a price and an allocation such that no trader has any incentive to trade and there is no excess demand of any good. While the problem was originally formulated by Walras [32] in 1874, the existence of such an equilibrium was established by Arrow and Debreu [2] in 1954 using a non-constructive fixed point argument. It turns out that the structure (and algorithmic properties) of market equilibrium problem depend on the properties of the aggregate demand and individual utility functions.

Tâtonnement. A natural approach to computing the equilibrium price (as originally envisaged by Walras) is an iterative algorithm termed as *tâtonnement* process where the prices of goods are updated locally as follows:

$$\pi_j^{t+1} = \pi_j^t + G_j(\pi_j^t, Z_j(\pi^t)) \tag{1}$$

where $\pi^t \in \Re^n_+$ is the price vector with components π_j^t, and $Z_j(\pi^t)$ the excess demand of good j at the t^{th} time step. G_j is a continuous function that preserves the sign of Z_j. *Tâtonnement* processes are especially important as they attempt to model the functioning of real markets. Such processes have traditionally been modeled in continuous domain using differential equations of the form:

$$\frac{d\pi_j}{dt} = G_j(Z_j(\pi))$$

e.g. by Samuleson [30]. Stability of these processes have been studied extensively in the literature [3, 25] (see [26] for a survey). It has been shown that if the utility functions satisfy the *weak gross substitute* (WGS) property then the continuous process is stable and converges to market equilibrium. The work on convergence of a discrete process of the form (1) has been limited. None of these works address the efficiency of computation in terms of rate of convergence or polynomial time computability. Experimental evidence using commercial packages to solve the market equilibrium problem suggests that in many cases, such tâtonnement processes converge very quickly to market equilibrium [5, 7]. However, polynomial time convergence of such a process was only recently established in exchange economies with WGS utilities by the works of Codenotti et al. [8]. Proving convergence of suitable tâtonnement processes for a wider class of utility functions (that do not necessarily satisfy the WGS property) remains an important open problem of interest to economists as well as computer scientists [19, 23, 8].

Convex Programming and other approaches. The convex programming approach has had more success in addressing a larger class of utility functions. In the Arrow-Debreu model polynomial-time computability has been established for CES utility functions (with $-\infty < \rho \leq 1$, $\rho \neq 0$) [6] and the class of utilities which satisfy *weak gross substitute* (WGS) property [3, 9]. Jain et al. [20] designed algorithms for computing market equilibria for homothetic, quasi-concave utilities in production. The linear utility functions is a special case for which the convex program designed by Primak et al.[28] and Jain [18] can be solved more efficiently using interior point methods [33]. In the Fisher model, Eisenberg and Gale [11] provide a convex program which solved the problem for linear and homogeneous utilities [10] (which include the utilities in resource allocation markets). These utilities are a special case of a more general class called monotone utilities [24] for which a polynomial time algorithm has been designed by Codenotti et al. [8]. Several combinatorial approaches, including primal-dual and auctions, have also been used for computing market equilibria [19, 15, 16].

Resource Allocation Markets. Of particular interest in a computer science context is the problem of pricing in resource allocation markets as formulated by Kelly [22]. In the special case of network-flow markets, traders (agents) wish to transmit flow between source-sink pairs and edges (links) are priced based on the demand and supply. In such markets the utilities of agents do not satisfy the WGS property and hence the existing tâtonnement-based techniques can no longer be applied. Recently Jain and Vazirani [19] have designed the first combinatorial strongly polynomial time algorithm to solve the pricing problem in the special case of network flows with single source and multiple sinks [19]. An important property of this approach is its combinatorial nature which may be given an economic interpretation.

1.1 Our Contributions

In this paper we design a new framework for solving the market equilibrium problem. This framework not only extends the class of utilities for which the market equilibrium can be computed in polynomial time, but also helps in designing an efficient tâtonnement process that converges close to market equilibrium.

A new formulation. We give a new formulation of the market equilibrium problem using *indirect utility function*. An indirect utility function $\widetilde{u}$ of price $\pi \in \Re^n_+$ and budget (or income) $e \in \Re_+$ gives the maximum utility achievable under those prices and budget as follows:

$$\widetilde{u}(\pi, e) = \max\{u(x) \mid x \in \Re^n_+, \pi \cdot x \leq e\}$$

where u is the utility function defined on allocation of goods. Although indirect utility functions have been extensively used in Economics to study the behavior of aggregate demand [24, 31], here we use them to formulate and solve the market equilibrium problem.

Polynomial time computation. Our formulation becomes a convex program if the indirect utility functions are convex on a suitably defined set of prices and income. This enables polynomial-time computation of (approximate) market equilibrium using standard convex programming techniques.

- We show that if the utility functions are quasi-linear (i.e., there is a good (say money) on which every trader has linear utility), then under mild concavity assumptions on the utility functions, the indirect utility functions are convex in prices. Thus, using our formulation, market equilibrium can be computed in polynomial time for a large class of utility functions in the Arrow-Debreu settings, for which no polynomial-time algorithm was known so far.

- In the Fisher setting, we show that indirect utility functions are convex if the utility functions are homogeneous of degree 1. Such utility functions include linear, Leontief, Cobb-Douglas, CES, resource allocation markets. If the utility function u is increasing in all its components, then a necessary and sufficient condition for convexity of the corresponding indirect utility function is (see Proposition 2.4 in [29]): $-\frac{x \cdot \partial^2 u(x) x}{\partial u(x) x} \leq 2$ for all x. Surprisingly, this condition has the same form as those for monotone utilities [8]. They turn out to be a special case of monotone utilities for which market equilibrium can be computed using ellipsoid method [8]. However, note that polynomial time convergent tâtonnement processes are not known for monotone utilities.

Efficient Tâtonnement Process. Our formulation enables us to design efficient tâtonnement processes that converge close to a market equilibrium in polynomial time whenever the indirect utility functions of traders are convex. Our algorithm mimics the natural dynamics for the markets as suggested by Walras: it iteratively adjusts a good's price upward when the demand for that good under current prices exceeds its supply; and downward when its supply exceeds its demand. This resolves the questions raised in [19, 23, 8] on convergence of tâtonnement processes for a very large class of utility functions.

The tâtonnement process, to obtain a $(1+\epsilon)$ approximate market equilibrium, requires every trader to perform at most $O(\epsilon^{-2} n \log n)$ computations of its demand. For multi-commodity flow resource allocation market, for example, the demand oracle is the shortest-path computation under the given edge-lengths (prices). Thus our algorithm needs $\tilde{O}(kn)$ shortest path computations for a market with k commodities and n edges. This contrasts against the algorithm of [19] for single-source multi-sink markets that needs $O(k^2)$ max-flow computations. We point out, however, that the algorithm of [19] computes an exact equilibrium while we compute only an approximate equilibrium.

Similar results were obtained by Fleischer et al. [13]; however they used convexity of *demand* functions as opposed to the indirect utility functions and hence could prove results only for a restricted sub-class of utility functions in Fisher economy.

The tâtonnement process is especially relevant for designing distributed algorithms in resource allocation markets [22]. For example, it shows that there is a congestion pricing scheme in networks that may be used to discover approximate market equilibrium prices. This may potentially be used to design congestion control [21] and routing algorithms [1] in networks. Similarly this may lead to insights on equilibrium pricing in pricing of "combinational goods" i.e. travel packages (airline tickets + hotel + car rentals) etc.

1.2 Organization

The rest of the paper is organized as follows. In Section 2 we define the market equilibrium problem and formulate a mathematical program using indirect utility functions. Section 2.1 gives some convex programming techniques for solving this formulation if the indirect utility functions are convex. Section 3 then presents the two prominent cases: quasi-linear utilities in exchange economies and several utilities in Fisher economy under which the indirect utility functions turn out to be convex. In Section 4, we present our tâtonnement algorithm for computing approximate market equilibria assuming convexity of indirect utility functions. Section 5 concludes with some open directions.

2 An Alternate Formulation using Indirect Utility Functions

We first describe the exchange market model. Let us consider m economic agents who represent traders of n goods. Let $\Re^n_+$ (resp. $\Re^n_{++}$) denote the subset of $\Re^n$ where the coordinates are non-negative (resp. strictly positive). The jth coordinate will stand for good j. Each trader i ($i = 1, \ldots, m$) is associated with

- a non-empty convex set $\mathcal{K}_i \subseteq \Re^n$ which is the set of all "feasible" allocations that trader i may receive (in many cases, $\mathcal{K}_i = \Re^n_+$),
- a *concave* utility function $u_i : \mathcal{K}_i \rightarrow \Re_+$ which represents her preferences for the different bundles of goods, and
- an initial endowment of goods $w_i = (w_{i1}, \ldots, w_{in})^\top \in \mathcal{K}_i$.

At given prices $\pi \in \Re^n_+$, the trader i sells her endowment, and gets the bundle of goods $x_i = (x_{i1}, \ldots, x_{in})^\top \in \mathcal{K}_i$ which maximizes $u_i(x)$ subject to budget constraint[1] $\pi \cdot x \leq \pi \cdot w_i$. A market equilibrium is a price vector $\pi \in \Re^n_+$ and bundles $x_i \in \mathcal{K}_i$ such that: $x_i \in \text{argmax}\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq \pi \cdot w_i\}$ for all i, and $\sum_i x_i \leq \sum_i w_i$. The above described market model is called an *exchange economy.*

We make the following standard assumption on the utility functions:

Assumption 2.1 *For $\pi \in \Re^n_+$, any $x_i \in$ argmax$\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq \pi \cdot w_i\}$ satisfies $\pi \cdot x_i = \pi \cdot w_i$.*

[1]For two vectors x and y, we use $x \cdot y$ to denote their inner product.

We now define a notion of indirect utility function induced by a utility function.

Definition 2.2 (Indirect utility function) *For trader i, the* indirect utility function $\widetilde{u}_i : \Re^n_+ \times \Re_+ \to \Re_+$ *gives the maximum utility achievable at given price and income:*

$$\widetilde{u}_i(\pi, e) = \max\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq e\}.$$

The following theorem characterizes the set of all equilibria.

Theorem 2.3 *The following program gives precisely the set of all market equilibria in the exchange economy.*

$$\begin{array}{rcll} \sum_i x_i & \leq & \sum_i w_i & \\ \widetilde{u}_i(\pi, \pi \cdot w_i) & \leq & u(x_i) & \textit{for all } i \\ \pi & \in & \Re^n_+ & \\ x_i & \in & \mathcal{K}_i & \textit{for all } i. \end{array} \tag{2}$$

Proof: From the definition, it follows that a market equilibrium satisfies the above inequalities. Now for converse, consider a solution $(\pi, x_1, \ldots, x_m)$ of the above program. From the second constraint and Assumption 2.1, it follows that $\pi \cdot x_i \geq \pi \cdot w_i$ for all i. Furthermore from the first constraint, it follows that $\sum_i \pi \cdot x_i \leq \sum_i \pi \cdot w_i$. This implies that $\pi \cdot x_i = \pi \cdot w_i$ for all i and hence the solution $(\pi, x_1, \ldots, x_m)$ is a market equilibrium. ∎

Note that the program (2) is convex when, for all i, the function $\widetilde{u}_i(\pi, \pi \cdot w_i)$ is a convex function of $\pi \in \Re^n_+$ and the utility function u_i is concave. Unfortunately, for many interesting utility functions u_i, the corresponding indirect utility function $\widetilde{u}_i$ is *not* convex. It turns out, however, that in *many* cases (as illustrated later in the paper), if we restrict the prices π to a convex set $\Pi \subseteq \Re^n_+$ that is guaranteed to contain an equilibrium price, the function $\widetilde{u}_i$ becomes convex in π. Therefore the program (2) reduces to the following convex program.

$$\begin{array}{rcll} \sum_i x_i & \leq & \sum_i w_i & \\ \widetilde{u}_i(\pi, \pi \cdot w_i) & \leq & u(x_i) & \text{for all } i \\ \pi & \in & \Pi & \\ x_i & \in & \mathcal{K}_i & \text{for all } i. \end{array} \tag{3}$$

In order to solve the above convex program using an ellipsoid algorithm, the convex set Π needs to be given in terms of a membership oracle.

2.1 Solving Program (3)

Assuming that the convex sets Π and $\mathcal{K}_i$ are bounded and full dimensional,[2] the convex program (3) can be solved to an arbitrary degree of precision by an ellipsoid-like algorithm using the evaluation oracle for the functions u_i and $\widetilde{u}_i$ and membership oracles for Π and $\mathcal{K}_i$. To outline this, we first quote a theorem from [17].

Lemma 2.4 (Grötschel-Lovász-Schrijver [17]-Thm. 4.3.13) *There exists an oracle-polynomial time algorithm that solves the following problem:*

Input: *A rational number $\epsilon > 0$, a centered d-dimensional convex body $(B; d, R, r, a_0)$ given by a weak membership oracle, and a convex function $f : \Re^d \to \Re$ given by an oracle that, for every $x \in \mathbb{Q}^d$ and $\delta > 0$, returns a rational number t such that $|f(x) - t| \leq \delta$.*

Output: *A vector $y \in S(B, \epsilon)$ such that $f(y) \leq f(x) + \epsilon$ for all $x \in S(B, -\epsilon)$.*

[2] The economies considered in this paper have unbounded Π and $\mathcal{K}_i$ in their description. However one can usually obtain bounds on the largest value that an allocation or a price can take. Moreover the cases that Π is not full dimensional can be handled using standard projection techniques.

A centered convex body B given as $(B; n, R, r, a_0)$ comes with a guarantee that $S(a_0, r) \subseteq B \subseteq S(\mathbf{0}, R)$ where $S(a, r)$ denotes a sphere of radius r centered at a. The variables in our convex program are $\pi, x_1, \ldots, x_m$. We let B to be the convex set defined by the linear constraints in the program (3) and $\pi \in \Pi$ and $x_i \in \mathcal{K}_i$. Assuming that Π and $\mathcal{K}_i$ are given as centered convex bodies, i.e., we know the inscribing and circumscribing spheres for Π and $\mathcal{K}_i$, we can find values of r, R, a_0 that satisfy the requirements. We let f to be the function $f(\pi, x_1, \ldots, x_m) = \max_i (\widetilde{u}_i(\pi, \pi \cdot w_i) - u_i(x_i))$.[3] Assuming the existence of evaluation oracles for u_i and $\widetilde{u}_i$, and membership oracles for Π and $\mathcal{K}_i$, we get an evaluation oracle for f. For the output, $S(B, \pm\epsilon)$ denotes the set of vectors which violate the constraints defining B by an additive term of $\pm\epsilon$ respectively. Thus we can solve (3) with an ϵ slack in the feasibility in oracle-polynomial time in the input, $\log(R/r)$, and $\log(1/\epsilon)$.

3 Convexity of the Indirect Utility Functions

In this section, we give a wide class of economies in which the indirect utility function $\widetilde{u}_i$ is convex in π over a set Π.

3.1 The Fisher Economy

The Fisher economy is a special case of the exchange economy when $\mathcal{K}_i = \Re^n_+$ and the endowments w_i of the traders are *proportional*, i.e.,

$$w_i = \lambda_i w$$

where $w \in \Re^n_{++}$ and $\lambda_1, \ldots, \lambda_m \in \Re_{++}$. In this case we let $\Pi = \{\pi \in \Re^n_+ \mid \pi \cdot w = 1\}$. Thus under any prices $\pi \in \Pi$, the income of trader i is fixed at λ_i.

We now quote a theorem of K.-H. Quah [29] which gives *necessary* and *sufficient* conditions on the utility functions u_i under which the indirect utility functions $\widetilde{u}_i(\pi, \lambda_i)$ are convex in $\pi \in \Pi$. We drop the subscript i to simplify the notation.

Theorem 3.1 (K.-H. Quah [29], Proposition 2.4) *Assume that the utility function $u : \Re^n_{++} \to \Re$ is continuous, quasi-concave, increasing in all arguments, and has the property that for any $\overline{x} \in \Re^n_{++}$, the set $\{x \in \Re^n_{++} \mid u(x) \geq u(\overline{x})\}$ is closed. Let $\lambda \in \Re_{++}$ be a constant.*

1. *Then, $\widetilde{u}(\pi, \lambda)$ is convex in prices π* if and only if *the functions μ_x are convex for all x, where $\mu_x : \Re_{++} \to \Re$ is defined by $\mu_x(s) = u(x/s)$.*

2. *Suppose, in addition, that u is $\mathcal{C}^2$, a twice differentiable function. Then μ_x is convex* if and only if *$\psi(x) = -\frac{x \cdot \partial^2 u(x) x}{\partial u(x) x} \leq 2$ for all x.*

An elementary proof of Theorem 3.1 is given, for completeness, in Appendix A.

Remark 3.2 *Contrast the condition $\psi(x) \leq 2$ above with the condition $\psi(x) < 4$ which is sufficient to guarantee that the induced demand function is* monotone *[8]. Recall that the demand function $x(\pi)$ is monotone if for any π, π', we have $(\pi - \pi') \cdot (x(\pi) - x(\pi')) \leq 0$. Thus if $\widetilde{u}$ is convex, the induced demand function is monotone.*

Corollary 3.3

1. *A concave homogeneous utility function u of degree α where $0 \leq \alpha \leq 1$, i.e., $u(sx) = s^\alpha u(x)$, results in convex indirect utility function $\widetilde{u}$ if u satisfies the conditions in Theorem 3.1.*

2. *If utility functions u_1 and u_2 satisfy the conditions in Theorem 3.1 and induce convex indirect utility functions, then so does $u_1 + u_2$.*

[3] It is enough to define f over B (and not entire $\Re^n_+$). See proof of Theorem 4.3.13 in [17] for more explanation.

Proof: For (1), note that $\mu_x(s) = s^{-\alpha}u(x)$ is a convex function of s. For (2), note that if $\mu_{1,x}$ and $\mu_{2,x}$ are convex functions then so is $\mu_{1,x} + \mu_{2,x}$. ■

Note, however, that some natural homogeneous utility functions of degree one (e.g., Leontief utilities and resource allocation utilities, defined later) do not satisfy the conditions in Theorem 3.1, in particular, the condition that the utility function is increasing in all arguments. However in the next theorem we show that the homogeneous utilities induce a convex indirect utility function even when they are not increasing in all arguments. The proof is given in Appendix A.

Theorem 3.4 *If the utility function $u : \Re^n_+ \to \Re$ is homogeneous (of degree one), i.e., $u(\alpha x) = \alpha u(x)$ for all $\alpha \in \Re_+$ and $x \in \Re^n_+$, then the indirect utility function $\widetilde{u}(\pi, \lambda)$ is convex in π for all $\lambda \in \Re_{++}$.*

The set of **homogeneous utility functions** of degree one includes the following well-studied utility functions. Here let $a \in \Re^n_+$. Linear utilities $u(x) = a \cdot x$, Leontief utilities $u(x) = \min_{j \in S} a_j x_j$ where $S \subseteq \{1, \ldots, n\}$, Cobb-Douglas utilities $u(x) = \prod_j x_j^{a_j}$ assuming $\sum_j a_j = 1$, CES utilities $u(x) = (\sum_j a_j x_j^{\rho})^{1/\rho}$ for $-\infty < \rho < 1$ and $\rho \neq 0$, and nested CES utilities [5, 7].

It also includes the **resource allocation utilities** defined as follows. Let k be a positive integer and let $A \in \Re^{n \times k}_+$ be a matrix and $c \in \Re^k_+$ be a vector. The resource allocation utility $u : \Re^n_+ \to \Re$ is defined as

$$u(x) = \max\{c \cdot y \mid y \in \Re^k_+, Ay \leq x\}. \tag{4}$$

The columns of matrix A can be thought of as "objects" that the trader wants to "build". A unit of an object l needs A_{jl} units of resource (or good) j and accrues c_l units of utility. The trader then builds y_l units of object l such that the total need for resources is at most x and the total utility $c \cdot y$ is maximized. This framework includes interesting markets like

1. Multi-commodity flow markets (in directed or undirected capacitated networks). Here trader i wants to send maximum amount of flow from source s_i to sink t_i such that the total cost of routing the flow under the prices π is at most her budget. The objects here are s_i-t_i paths and the resources are the edges.

2. Steiner-tree markets in undirected (resp. directed) capacitated networks. Here trader i is associated with a subset S_i of nodes and wants to build maximum fractional packing of Steiner trees connecting S_i (resp. fractional arborescences rooted at some $r_i \in S_i$ connecting S_i to r_i) such that the total cost of building under the prices π is at most her budget. The objects here are Steiner trees (resp. arborescences). Note that computing a profit maximizing demand in undirected Steiner-tree market is NP-hard. Therefore the running times of the algorithms are only oracle-polynomial.

From Corollary 3.3, the **additive separable concave** utilities also induce a convex indirect utility functions: (1) $u(x_1, \ldots, x_n) = \sum_j a_j x_j^{\rho_j}$ where $a_j \in \Re_{++}$ and $0 \leq \rho_j \leq 1$; (2) $u(x_1, \ldots, x_n) = \sum_j \log(1 + a_j x_j)$ where $a_j \in \Re_{++}$ [4] — follows from the fact that $\log(1 + \frac{a_j x_j}{s})$ is a convex function of s.

3.2 Quasi-linear Utilities in Exchange Economies

In this section, we consider the exchange economy with quasi-linear utility functions given as follows. Let $\mathcal{K}_i = \Re \times \Re^{n-1}_+$ for all i. There is a distinguished good, say good 1, for which every trader has linear utility. Thus the utility function of trader i is given by

$$u_i(x_{i1}, x_{i2}, \ldots, x_{in}) = a_i x_{i1} + v_i(x_{i2}, \ldots, x_{in}) \tag{5}$$

where $a_i \in \Re_{++}$ and $v_i : \Re^{n-1}_+ \to \Re_+$ is an *arbitrary* concave function. Consider the set $\Pi \in \Re^n_+$ of prices defined as

$$\Pi = \{\pi \in \Re^n_+ \mid \pi_1 = 1\}.$$

Lemma 3.5 *The indirect utility function $\widetilde{u}_i(\pi, \pi \cdot w_i)$ is a convex function of π if π is restricted to Π.*

Proof: Since good 1 has a positive linear utility, it is clear that $x_i^* = \text{argmax}\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq \pi \cdot w_i\}$ satisfies $\pi \cdot x_i^* = \pi \cdot w_i$ and Assumption 2.1 is satisfied. Since $\pi_1 = 1$, we get that $x_{i1}^* = \pi \cdot w_i - \sum_{j \neq 1} \pi_j x_{ij}^*$. Substituting this value into the expression for u_i we get that

$$\widetilde{u}_i(\pi, \pi \cdot w_i) = a_i(\pi \cdot w_i - \sum_{j \neq 1} \pi_j x_{ij}^*) + v_i(x_{i2}^*, \ldots, x_{in}^*).$$

Thus in fact

$$\widetilde{u}_i(\pi, \pi \cdot w_i) = \max_{x_i \in \mathcal{K}_i} \left(a_i(\pi \cdot w_i - \sum_{j \neq 1} \pi_j x_{ij}) + v_i(x_{i2}, \ldots, x_{in}) \right).$$

For a fixed $x_i \in \mathcal{K}_i$, the expression in the bracket on the right-hand-side is a linear (and hence, convex) function of π. Thus $\widetilde{u}_i$ is a pointwise maximum of a family of convex functions in π, and is convex. ∎

4 A Tâtonnement Algorithm for Solving (3)

We first define a notion of an approximate market equilibrium for the exchange economy and present a tâtonnement process that converges to an approximate market equilibrium whenever the indirect utility function $\widetilde{u}_i(\pi, \pi \cdot w_i)$ for each trader i is convex in prices π restricted to $\pi \in \Pi$. For some technical reason, we assume that the set Π satisfies the following property: for any vector $p \in \Re_+^n$, there exists $\alpha \in \Re_{++}$ such that $\alpha p \in \Pi$. Note that this requirement is satisfied by the sets Π for quasi-linear utilities and the utilities in Fisher markets.

We consider the following notion of weak approximate market equilibrium. To simplify the definition, we assume that $\mathcal{K}_i \subseteq \Re_+^n$, i.e., we let x_{ij} take only non-negative values.

Definition 4.1 (Weak $(1+\epsilon)$-approximate market equilibrium) *A price vector $\pi \in \Pi$ and bundles $x_i \in \mathcal{K}_i$ for each trader i are called a weak $(1+\epsilon)$-approximate market equilibrium in the exchange economy if*

1. *The utility of x_i to trader i is at least that of the utility-maximizing bundle under prices π: $u_i(x_i) \geq \widetilde{u}_i(\pi, \pi \cdot w_i)$ for each i,*
2. *The total demand is at most $(1+\epsilon)$ times the supply: $\sum_i x_i \leq (1+\epsilon) \sum_i w_i$, and*
3. *The market clears: $\pi \cdot \sum_i w_i \leq \pi \cdot \sum_i x_i$.*

If however $\mathcal{K}_i \not\subseteq \Re_+^n$, we use a standard technique of "shifting" the space so that x_{ij} are non-negative. This, however, needs that $\mathcal{K}_i$ is bounded below and we know these bounds. It also weakens the notion of approximate market equilibrium and we omit the details from this extended abstract. Shifting has also been used to address similar problems arising while solving linear programs with negative entries [27].

The main result of this section is summarized in the following theorem.

Theorem 4.2 *There exists a tâtonnement algorithm that computes a weak $(1+\epsilon)$-approximate market equilibrium in the exchange economy for which a set Π containing an equilibrium price is known such that for each i, the indirect utility function $\widetilde{u}_i(\pi, \pi \cdot w_i)$ is a convex function of π when restricted to $\pi \in \Pi$.*

In the algorithm, each trader i makes $O(\epsilon^{-2} n \log n)$ calls to her "demand" oracle: given prices $\pi \in \Pi$, compute $x_i \in$ argmax$\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq \pi \cdot w_i\}$.

1. Initialize $p_j = 1$ for $1 \le j \le n$.
2. Repeat for $N = \frac{n}{\delta} \log_{1+\delta} n$ iterations:
   (a) Find $\alpha > 0$ such that $\alpha p \in \Pi$. Announce prices $\pi = \alpha p$.
   (b) Each trader i computes $x_i \in \text{argmax}\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \le \pi \cdot w_i\}$.
   (c) Compute the aggregate demand $X = \sum_i x_i$ and let $\sigma = \frac{1}{\max_j X_j}$ where X_j denotes the aggregate demand of good j.
   (d) Update for each good j:
   $$p_j \leftarrow p_j \left(1 + \delta\sigma X_j\right).$$
3. Output for each i:
   $$\overline{x}_i = \frac{\sum_{r=1}^{N} \sigma(r) x_i(r)}{\sum_{r=1}^{N} \sigma(r)}$$
   where $x_i(r)$ and $\sigma(r)$ are the values of x_i and σ in the rth iteration.
4. Output
   $$\overline{\pi} = \frac{\sum_{r=1}^{N} \sigma(r) \pi(r)}{\sum_{r=1}^{N} \sigma(r)}$$
   where $\pi(r)$ and $\sigma(r)$ are the values of $\hat{\pi}$ and σ in the rth iteration.

Figure 1: Algorithm for the convex program (3)

4.1 The Algorithm

Without loss of generality, we scale the endowments w_i so that $\sum_i w_i = \mathbf{1}$, the vector of all ones. This also implies that we scale the vectors in $\mathcal{K}_i$. We emphasize that the algorithm also works without scaling; however the scaling simplifies the presentation. The algorithm is given in Figure 1. Here $\delta > 0$ is a constant to be fixed later. The algorithm goes in N iterations. In each iteration, we first scale the current price vector p so that it lies in Π. We then "announce" this price vector and receive in response the utility-maximizing bundles $x_i \in \mathcal{K}_i$. We then update the price vector p according to the aggregate demands X_j of goods j as given in Step 2d.

Note that this update is essentially same as (within a $(1+\delta)$ factor) to the following natural update in terms of *excess demand*. Let $Z_j = X_j - \sum_i w_{ij} = X_j - 1$ be the excess demand of good j. We can update p as:

$$p_j \leftarrow p_j(1 + \delta\sigma Z_j).$$

This is so because $(1 + \delta\sigma Z_j) \approx (1 + \delta\sigma X_j)(1 - \delta\sigma)$, which is in turn true since $Z_j = X_j - 1$ and $\delta\sigma$ is small. The extra factor $(1 - \delta\sigma)$ is common to all goods j and is factored away in the price scaling step.

The algorithm in the end outputs, $\overline{\pi}$ and $\overline{x}_i$ for all i, the weighted average of the prices and allocations computed in N iterations.

Lemma 4.3 *The outputs $\overline{x}_i$ and $\overline{\pi}$ satisfy $u_i(\overline{x}_i) \ge \widetilde{u}_i(\overline{\pi}, \overline{\pi} \cdot w_i)$ for each i.*

Proof: Since $\widetilde{u}_i(\pi, \pi \cdot w_i)$ is convex when $\pi \in \Pi$ and $u_i(x_i)$ is concave when $x_i \in \mathcal{K}_i$, we have

$$\widetilde{u}_i(\overline{\pi}, \overline{\pi} \cdot w_i) \le \frac{\sum_r \sigma(r) \widetilde{u}_i(\pi(r), \pi(r) \cdot w_i)}{\sum_r \sigma(r)} = \frac{\sum_r \sigma(r) u_i(x_i(r))}{\sum_r \sigma(r)} \le u_i(\overline{x}_i).$$

■

The following main lemma about the output is proved below The proof is based on the standard application of "experts theorem" or "multiplicative update" technique used previously in solving packing and covering linear programs [27, 14, 12].

Lemma 4.4 *The outputs $\overline{x}_i$ satisfy $\sum_i \overline{x}_i \le \frac{1}{1-2\delta} \sum_i w_i$.*

We set $\delta = \frac{\epsilon}{2(1+\epsilon)}$ so that $\frac{1}{1-2\delta} = 1+\epsilon$. The proof of Theorem 4.2 now follows from Lemmas 4.3, 4.4, and Assumption 2.1 on the utility functions.

Proof of Lemma 4.4. Let $\overline{x} = \sum_i \overline{x}_i$ and let $(\overline{x})_j$ denote the jth coordinate of $\overline{x}$. To this end, let us define a potential

$$\Phi(r) = \sum_j p_j(r) \tag{6}$$

where $p_i(r)$ denote the value of p_i in the beginning of rth iteration. From the step 2d in the algorithm, we have

$$\Phi(r+1) = \Phi(r) + \delta\sigma(r) \sum_j p_j(r) X_j(r)$$

where $X_j(r)$ denotes the value of X_j in the rth iteration. Thus

$$\begin{aligned} \frac{\Phi(r+1)}{\Phi(r)} &= 1 + \delta\sigma(r) \sum_j \frac{p_j(r)}{\Phi(r)} X_j(r) \\ &= 1 + \delta\sigma(r) \\ &\le \exp(\delta\sigma(r)). \end{aligned}$$

The second equality follows from the fact that $\sum_j p_j(r) X_j(r) = \frac{1}{\alpha(r)} \sum_j \pi_j(r) X_j(r)$ which is, by Assumption 2.1, equal to $\frac{1}{\alpha(r)} \sum_j \pi_j(r) \sum_i w_{ij} = \frac{1}{\alpha(r)} \sum_j \pi_j(r) = \sum_j p_j(r) = \Phi(r)$. Here $\alpha(r)$ is the value of α in rth iteration.

Thus after taking telescoping product, we get

$$\Phi(N+1) \le \Phi(1) \cdot \exp\left(\delta \sum_r \sigma(r)\right) = n \cdot \exp\left(\delta \sum_r \sigma(r)\right). \tag{7}$$

On the other hand, observe that

$$\begin{aligned} \Phi(N+1) = \sum_j p_j(N+1) &= \sum_j \prod_{r=1}^{N} (1 + \delta\sigma(r) X_j(r)) \\ &\ge \sum_j \exp\left(\delta(1-\delta) \sum_r \sigma(r) X_j(r)\right) \\ &\ge \max_j \exp\left(\delta(1-\delta) \sum_r \sigma(r) X_j(r)\right) \\ &= \exp\left(\delta(1-\delta) \max_j \sum_r \sigma(r) X_j(r)\right). \end{aligned}$$

The first inequality follows from the elementary fact that $1 + \mu \ge \exp(\mu(1-\delta))$ for all $0 < \mu < \delta < \frac{1}{2}$. Combining the above observation with (7), we get

$$\delta(1-\delta) \max_j \sum_r \sigma(r) X_j(r) \le \log \Phi(N+1) \le \log n + \delta \sum_r \sigma(r).$$

Therefore,

$$\max_j (\overline{x})_j = \max_j \frac{\sum_r \sigma(r) X_j(r)}{\sum_r \sigma(r)} \tag{8}$$

$$\leq \frac{1}{1-\delta} + \left(\frac{\log n}{\delta(1-\delta)\sum_r \sigma(r)} \right). \tag{9}$$

Now we "charge" the second term on the right-hand-side in (9) to $\max_j (\overline{x})_j$ as follows. Note that at least one p_j increases by a factor $(1+\delta)$ in any iteration. Thus after $N = \frac{n}{\delta} \log_{1+\delta} n$ iterations, $\max_j p_j(N+1) \geq n^{1/\delta}$. Also

$$(\overline{x})_j = \frac{\sum_r \sigma(r) X_j(r)}{\sum_r \sigma(r)} \geq \frac{\log p_j(N+1)}{\delta \sum_r \sigma(r)}.$$

Thus $\max_j (\overline{x})_j \geq \frac{\log n}{\delta^2 \sum_r \sigma(r)}$. Putting all pieces together, we get

$$\max_j (\overline{x})_j \leq \frac{1}{1-\delta} + \left(\frac{\delta \max_j (\overline{x})_j}{1-\delta} \right).$$

Thus $\max_j (\overline{x})_j \leq \frac{1}{1-2\delta}$.

5 Conclusions and Future Work

In this paper, we give an alternate formulation of the market equilibrium problem using indirect utility functions. This formulation becomes a convex program (which can be solved using standard techniques), if the indirect utility functions of traders are convex.

In Fisher economies, the most commonly studied utility functions such as linear, Leontiff, Cobb-Douglas, CES and even resource allocation markets induce convex indirect utility functions. For exchange economies, the indirect utility function becomes convex if the utilities are quasi-linear. This expands the class of utility functions for which (approximate) market equilibrium can be computed in polynomial time. We also give a discrete tâtonnement process that converges to a weak approximate market equilibrium in these cases.

One of the open question is to precisely characterize the class of utility functions in the exchange economies for which the indirect utility function becomes convex. We suspect that this class is larger than quasi-linear utilities. We feel that there is a way to incorporate linear utilities in the exchange model in our framework. In fact, it should also be possible to incorporate production economies in the above framework as well.

Our definitions of approximate market equilibrium is weak because the budget constraints of traders are satisfied only in the aggregate sense. Some of the traders may be spending significantly more than their budget. Moreover, some positively priced items may not be fully allocated. A notion of strongly approximate market equilibrium may be defined on the lines of [15], where budget constraints of no trader may exceed by a factor more than $(1+\epsilon)$ and no item with positive price is under-demanded. Under this definition it might be possible to prove the "closeness" of the discovered prices to the equilibrium prices (see e.g., [16]). If we set $\delta = O(\frac{\epsilon \min_i \lambda_i}{\sum_i \lambda_i})$, where λ_i is the income of trader i in a Fisher economy, our tâtonnement algorithm obtains a strong $(1+\epsilon)$ approximate market equilibrium in the above sense in $O((\frac{\epsilon \min_i \lambda_i}{\sum_i \lambda_i})^{-2} n \log n)$ iterations. It will be very interesting to develop a tâtonnement algorithm that converges to a strong approximate market equilibrium in near linear number of iterations.

Finally, it is interesting to note that the continuous time version of our tâtonnement process can be described as

$$\frac{d\pi_j}{dt} = \pi_j Z_j$$

where $Z_j = \sum_j x_{ij} - \sum_i w_{ij}$ is the excess demand of good j. Under what conditions is this process or its "time-average" $\hat{\pi}_j = \frac{1}{t}\int_{\tau=0}^{t} \pi_j d\tau$ stable and does converge to the equilibrium?

6 Acknowledgements

We thank Amin Saberi and Lisa Fleischer for many useful discussions related to this work.

References

[1] E. Altman, T. Boulogne, R. Azouzi, and T. Jimenez. A survey on networking games in telecommunications. *Computers and Operations Research*, 2004.

[2] K. Arrow and G. Debreu. Existence of an Equilibrium for a Competitive Economy. *Econometrica*, 22:265–290, 1954.

[3] K.J. Arrow, H.D. Black, and L. Hurwicz. On the Stability of the Competetive Equilibrium, II. *Economitrica*, 27:82–109, 1959.

[4] N. Chen, Xiatie Deng, Xiaoming Sun, and Andrew Yao. Fisher Equilibrium Price with a class of Concave Utility Functions. In *Proceedings of ESA 2004*, 2004.

[5] Bruno Codenotti, Benton McCune, Sriram V. Pemmaraju, Rajiv Raman, and Kasturi Varadarajan. An experimental study of different approaches to solve the market equilibrium problem. In *ALENEX/ANALCO*, pages 167–179, 2005.

[6] Bruno Codenotti, Benton McCune, Sriram Penumatcha, and Kasturi R. Varadarajan. Market equilibrium for ces exchange economies: Existence, multiplicity, and computation. In *FSTTCS*, pages 505–516, 2005.

[7] Bruno Codenotti, Benton McCune, Rajiv Raman, and Kasturi Varadarajan. Computing equilibrium prices: Does theory meet practice? In *Proceedings of ESA 2005*, 2005.

[8] Bruno Codenotti, Benton McCune, and Kasturi Varadarajan. Market equilibrium via the excess demand function. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 74–83, New York, NY, USA, 2005. ACM Press.

[9] Bruno Codenotti, Sriram Pemmaraju, and Kasturi Varadarajan. On the polynomial time computation of equilibria for certain exchange economies. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 72–81, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[10] E. Eisenberg. Aggregation of utility functions. *Management Sciences*, 7(4):337–350, 1961.

[11] E. Eisenberg and D. Gale. Consensus of Subjective Probabilities: The Pari-Mutuel Method. *Annals of Mathematical Statistics*, 30:165–168, 1959.

[12] L. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discrete Math.*, 13:505–520, 2000.

[13] L. Fleischer, R. Khandekar, and A. Saberi. A fast and distributed algorithm for computing fisher equilibrium in economies with decreasing marginal returns. Manuscript, 2005.

[14] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[15] Rahul Garg and Sanjiv Kapoor. Auction algorithms for market equilibrium. *Math. Oper. Res.*, 31(4):714–729, 2006.

[16] Rahul Garg and Sanjiv Kapoor. Price roll-backs and path auctions: An approximation scheme for computing the market equilibrium. In *WINE*, pages 225–238, 2006.

[17] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, 1988.

[18] Kamal Jain. A Polynomial Time Algorithm for Computing the Arrow-Debreau Market equilibrium for Linear Utilities. FOCS 2004.

[19] Kamal Jain and Vijay Vazirani. Eisenberg-gale markets: Algorithms and structural properties. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, 2007.

[20] Kamal Jain, Vijay V. Vazirani, and Yinyu Ye. Market equilibria for homothetic, quasi-concave utilities and economies of scale in production. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 63–71, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[21] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[22] Frank Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[23] Frank P. Kelly and Vijay Vazirani. Rate control as a market equilibrium. Manuscript, 2002.

[24] Andrew Mas-Colell. *The Theory of General Economic Equilibrium: A Differential Approach.* Cambridge University Press, Cambridge, 1985.

[25] Takashi Negishi. A note on the stability of an economy where all goods are gross substitutes. *Econometrica*, 26(3):445–447, 1958.

[26] Takashi Negishi. The stability of a competitive economy: A survey article. *Econometrica*, 30(4):635–669, 1962.

[27] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20:257–301, 1995.

[28] M.E. Primak. A Converging Algorithm for a Linear Exchange Model. *Applied Mathematics and Computations*, 52:223–231, 1992.

[29] John K.-H. Quah. The monotonicity of individual and market demand. *Econometrica*, 68(4):911–930, 2000.

[30] P. Samuelson. *Foundations of Economic Analysis.* Harvard University Press, Boston, 1947.

[31] H. Varian. *Microeconomic Analysis.* W. W. Norton, New York, 1992.

[32] L. Walras. Elements of Pure Economics, or the Theory of Social Wealth (in French). Lausanne, Paris, 1874.

[33] Yinyu Ye. A path to the arrow-debreu competitive market equilibrium. *Mathematical Programming*, 2006.

A Additional proofs

Proof of Theorem 3.1. The assumptions on the utility function imply that for any $x \in \Re^n_{++}$ there is a supporting price π such that $x \in \operatorname{argmax}\{u(x) \mid x \in \Re^n_{++}, \pi \cdot x \leq \lambda\}$.

To prove (1), we first assume that $\widetilde{u}$ is convex in π and show that μ_x is convex. Let π_1 and π_2 be supporting prices of x/s_1 and x/s_2 respectively, where s_1 and s_2 are two positive numbers. Thus $\mu_x(s_1) = \widetilde{u}(\pi_1, \lambda)$ and $\mu(s_2) = \widetilde{u}(\pi_2, \lambda)$. Then we have

$$\begin{aligned}
\alpha\mu_x(s_1) + (1-\alpha)\mu_x(s_2) &= \alpha\widetilde{u}(\pi_1, \lambda) + (1-\alpha)\widetilde{u}(\pi_2, \lambda) \\
&\geq \widetilde{u}(\alpha\pi_1 + (1-\alpha)\pi_2, \lambda) \\
&\geq \mu_x(\alpha s_1 + (1-\alpha)s_2).
\end{aligned}$$

The first inequality follows from the convexity of $\widetilde{u}$ with respect to prices; the second inequality follows from the fact that $\mu_x(\alpha s_1 + (1-\alpha)s_2) = u(x/[\alpha s_1 + (1-\alpha)s_2])$, and

$$(\alpha\pi_1 + (1-\alpha)\pi_2) \cdot \left(\frac{x}{\alpha s_1 + (1-\alpha)s_2}\right) = \lambda.$$

Now we assume that μ_x is convex for all x and show that $\widetilde{u}$ is convex in prices. Let π_1 and π_2 be two prices and let $x \in \operatorname{argmax}\{u(x) \mid x \in \Re^n_{+}, (\alpha\pi_1 + (1-\alpha)\pi_2) \cdot x \leq \lambda\}$. Since μ_x is convex and $(\alpha\pi_1 + (1-\alpha)\pi_2) \cdot x = \lambda$, we have $u(x) = \mu_x(1) \leq \alpha\mu_x\left(\frac{\pi_1 \cdot x}{\lambda}\right) + (1-\alpha)\mu_x\left(\frac{\pi_2 \cdot x}{\lambda}\right)$. Therefore,

$$\begin{aligned}
\widetilde{u}(\alpha\pi_1 + (1-\alpha)\pi_2, \lambda) &= u(x) \leq \alpha\mu_x\left(\frac{\pi_1 \cdot x}{\lambda}\right) + (1-\alpha)\mu_x\left(\frac{\pi_2 \cdot x}{\lambda}\right) \\
&= \alpha u\left(\frac{\lambda x}{\pi_1 \cdot x}\right) + (1-\alpha)u\left(\frac{\lambda x}{\pi_2 \cdot x}\right) \\
&\leq \alpha\widetilde{u}(\pi_1, \lambda) + (1-\alpha)\widetilde{u}(\pi_2, \lambda)
\end{aligned}$$

where the last inequality follows from the definition of $\widetilde{u}$.

The prove (2), we need only check that $\mu''_x(s) = \frac{x}{s^2} \cdot \partial^2 u\left(\frac{x}{s}\right)\frac{x}{s^2} + \partial u\left(\frac{x}{s}\right) \cdot \frac{x}{s^3}$. This is positive if and only if $\psi\left(\frac{x}{s}\right) \leq 2$.

Proof of Theorem 3.4. Let price vectors $\pi, \pi_1, \pi_2 \in \Re^n_+$ satisfy $\pi = \alpha\pi_1 + (1-\alpha)\pi_2$ for some $0 \leq \alpha \leq 1$. Let $x \in \Re^n_+$ be such that $\pi \cdot x = \lambda$ and $u(x) = \widetilde{u}(\pi, \lambda)$. Define $x_1 = \frac{\lambda x}{\pi_1 \cdot x}$ and $x_2 = \frac{\lambda x}{\pi_2 \cdot x}$. Note that $\pi_1 \cdot x_1 = \pi_2 \cdot x_2 = \lambda$ and hence $\widetilde{u}(\pi_1, \lambda) \geq u(x_1)$ and $\widetilde{u}(\pi_2, \lambda) \geq u(x_2)$. Using the homogeneity of u, we also get that $u(x) = \frac{\pi_1 \cdot x}{\lambda}u(x_1) \leq \frac{\pi_1 \cdot x}{\lambda}\widetilde{u}(\pi_1, \lambda)$ and $u(x) = \frac{\pi_2 \cdot x}{\lambda}u(x_2) \leq \frac{\pi_2 \cdot x}{\lambda}\widetilde{u}(\pi_2, \lambda)$.

Note that $\alpha(\pi_1 \cdot x) + (1-\alpha)(\pi_2 \cdot x) = \lambda$. Now

$$\begin{aligned}
\left(\frac{\alpha\lambda}{\pi_1 \cdot x} + \frac{\lambda(1-\alpha)}{\pi_2 \cdot x}\right) &= \left(\frac{\alpha\lambda}{\pi_1 \cdot x} + \frac{\lambda(1-\alpha)}{\pi_2 \cdot x}\right)\left(\frac{\alpha(\pi_1 \cdot x)}{\lambda} + \frac{(1-\alpha)(\pi_2 \cdot x)}{\lambda}\right) \\
&= \alpha^2 + \alpha(1-\alpha)\left(\frac{\pi_1 \cdot x}{\pi_2 \cdot x} + \frac{\pi_2 \cdot x}{\pi_1 \cdot x}\right) + (1-\alpha)^2 \\
&\geq \alpha^2 + 2\alpha(1-\alpha) + (1-\alpha)^2 \\
&= 1.
\end{aligned}$$

To complete the proof we now observe

$$\begin{aligned}
\widetilde{u}(\pi, \lambda) = u(x) &\leq u(x)\left(\frac{\alpha\lambda}{\pi_1 \cdot x} + \frac{\lambda(1-\alpha)}{\pi_2 \cdot x}\right) \\
&\leq \left(\frac{\pi_1 \cdot x}{\lambda}\widetilde{u}(\pi_1, \lambda)\right)\frac{\alpha\lambda}{\pi_1 \cdot x} + + \left(\frac{\pi_2 \cdot x}{\lambda}\widetilde{u}(\pi_2, \lambda)\right)\frac{\lambda(1-\alpha)}{\pi_2 \cdot x} \\
&= \alpha\widetilde{u}(\pi_1, \lambda) + (1-\alpha)\widetilde{u}(\pi_2, \lambda).
\end{aligned}$$