



A combinatorial polynomial algorithm for the linear Arrow–Debreu market[☆]

Ran Duan, Kurt Mehlhorn^{*}

Max-Planck-Institut für Informatik, Saarbrücken, Germany

ARTICLE INFO

Article history:

Received 30 August 2013

Available online 12 December 2014

ABSTRACT

We present the first combinatorial polynomial time algorithm for computing the equilibrium of the Arrow–Debreu market model with linear utilities. Our algorithm views the allocation of money as flows and iteratively improves the balanced flow as in [11] for Fisher’s model. We develop new methods to carefully deal with the flows and surpluses during price adjustments. Our algorithm performs $O(n^6 \log(nU))$ maximum flow computations, where n is the number of agents and U is the maximum integer utility. The flows have to be presented as numbers of bitlength $O(n \log(nU))$ to guarantee an exact solution. Previously, [22,29] have given polynomial time algorithms for this problem, which are based on solving convex programs using the ellipsoid algorithm and the interior-point method, respectively.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

We provide the first combinatorial polynomial algorithm for computing the equilibrium of the linear version of an economic market model formulated by Walras¹ in 1874 [28]. The model is also known as the linear exchange model. In this model, every agent has an initial endowment of some goods and a utility function over sets of goods. Goods are assumed to be divisible. The market clears at a set of prices if each agent can spend its entire budget (= the total value of its goods at the set of prices) on a bundle of goods with maximum utility, and all goods are completely sold. Market clearing prices are also called equilibrium prices. In the linear version of the problem, all utility functions are linear.

Formally, the linear model is defined as follows. We may assume w.l.o.g. that the number of goods is equal to the number of agents and that the i -th agent owns the i -th good. Let $u_{ij} \geq 0$ be the utility for buyer i if all of good j is allocated to him. If fraction x_{ij} of good j is allocated to buyer i , the total utility for i is

$$\sum_j u_{ij} x_{ij}.$$

We make the standard assumption that each agent likes some good, i.e., for all i , $\max_j u_{ij} > 0$, and that each good is liked by some agent, i.e., for all j , $\max_i u_{ij} > 0$. Agents are selfish and spend money only on goods that give them maximum utility per unit of money, i.e., if $p = (p_1, \dots, p_n)$ is a price vector then buyer i is only willing to buy goods j with $u_{ij}/p_j = \max_j u_{ij}/p_j$. An *equilibrium* is a vector p of positive prices and allocations $x_{ij} \geq 0$ such that

[☆] A preliminary version of this paper was presented at ICALP 2013.

^{*} Corresponding author.

E-mail addresses: duanran@mpi-inf.mpg.de (R. Duan), mehlhorn@mpi-inf.mpg.de (K. Mehlhorn).

¹ Walras’ model is actually more general and also involves production.

- all goods are completely sold: $\sum_i x_{ij} = 1$
- all money is spent: $\sum_j x_{ij} p_j = p_i$
- only maximum utility per unit of money goods are bought:

$$x_{ij} > 0 \Rightarrow \frac{u_{ij}}{p_j} = \alpha_i, \quad \text{where } \alpha_i = \max_{\ell} \frac{u_{i\ell}}{p_{\ell}}$$

The existence of an equilibrium is non-obvious. Walras argued existence algorithmically. Assume that a vector p of prices is not an equilibrium vector. Then there is a good for which demand exceeds supply. Increase the price of the good until demand equals supply. Of course, this might destroy the equilibrium for other goods. However, the effect on the other goods will be smaller (he did not define this term) and hence the process will converge.

Fisher [15] introduced a somewhat simpler model (agents come with budgets) in 1891 and showed how to compute an equilibrium for the case of three agents and three goods by an analog computer. The computer is a hydro-mechanical device that stabilizes at equilibrium prices and allocations [3].

The first rigorous existence proof is due to Wald [27]. It required fairly strong assumptions. The existence proof for the general model was given by Arrow and Debreu [2] in 1954. They proved that the market equilibrium always exists if the utility functions are concave. The result is prominently mentioned in their Nobel prize laudation and the market is usually referred to as the “Arrow–Debreu market”. However, their proof is based on Kakutani’s fixed-point theorem and hence non-constructive. Gale [16,17] gives necessary and sufficient conditions for the existence of an equilibrium in the linear model; see Section 2.5.

The development of algorithms started in the 60s. There is a wide literature in the economics and mathematics community, and since 2000 also in the algorithm community. The survey paper by Codenotti, Pemmaraju and Varadarajan [8] surveys the literature until 2004. Early algorithms, for example by Scarf, Smale, Kuhn, and Todd, are inspired by fixed-point proofs or are Newton-based and compute approximations. The running time of these algorithms is exponential.

First approximation algorithms with polynomial running time appeared after 2000. Jain, Madhian, and Saberi [23] gave a polynomial time approximation scheme. Devanur and Vazirani [12] obtained a strongly polynomial time approximation scheme and Garg and Kapoor [19] gave a simplified approximation scheme. Recently, Ghyasvand and Orlin [21] gave an approximation scheme with running time $O(\frac{n}{\epsilon}(m + n \log n))$; here m is the number of positive utilities u_{ij} . The approximation algorithms are combinatorial.

Exact algorithms are also known. Eaves [13] presented the first exact algorithm for the linear exchange model. He reduced the model to a linear complementary problem which is then solved by Lemke’s algorithm. The algorithm is not polynomial time. Garg, Mehta, Sohoni, and Vazirani [18] give a combinatorial interpretation of the algorithm. Polynomial time exact algorithms were obtained based on the characterization of equilibria as the solution set of a convex program. The recent paper by Devanur, Garg, and Végh [10] surveys these characterizations. For example, Nenakov and Primak [24] showed that equilibria are precisely the solutions of the system

$$p_i > 0 \quad x_{ij} \geq 0 \quad \sum_j u_{ij} x_{ij} \geq \frac{u_{ik}}{p_k} p_i \quad \text{for all } i \text{ and } k.$$

Note that $\frac{u_{ik}}{p_k} p_i$ is the utility obtained by the i -th buyer if he spends his entire budget, which is p_i under the assumption that his good is completely sold, on good k . The program above is not convex in the variables p_i and x_{ij} . However, after the substitution $p_i = e^{\pi_i}$ it becomes a convex program in the variables π_i and x_{ij} . Jain [22] rediscovered this convex program and showed how to solve it with a nontrivial extension of the ellipsoid method. His algorithm is the first polynomial time algorithm for the linear Arrow–Debreu market. Ye [29] showed that polynomial time can also be achieved with the interior point method. The latter algorithm runs in time $O(n^4 \log U)$ for integral utilities bounded by U and is significantly faster than the algorithm presented in this paper.

In Fisher markets each agent comes with a budget to the market. It is a special case of the Arrow–Debreu market and algorithmically simpler. Eisenberg and Gale [14] characterized equilibria by a convex program. With the advent of the ellipsoid method, their characterization gave rise to a polynomial algorithm. The first combinatorial algorithm was given by Devanur, Padimitriou, Saberi, and Vazirani [11]. Their algorithm uses a maximum flow algorithm as a black box. When the input data are integral, their algorithm needs $O(n^5 \log U + n^4 \log e_{\max})$ max-flow computations, where n is the number of buyers, U the largest integer utility, and $e_{\max}$ the largest initial amount of money of a buyer. If we use the common $O(n^3)$ max-flow algorithm (see [1]), their running time is $O(n^8 \log U + n^7 \log e_{\max})$. Orlin [25] improved the running time to $O(n^4 \log U + n^3 \log e_{\max})$ and also gave the first strongly polynomial algorithm with running time $O(n^4 \log n)$. Our work is inspired by these papers.

Our algorithm has the advantage of being combinatorial (see Fig. 2 for a complete listing), and hence, gives additional insight in the nature of the problem. We obtain equilibrium prices by a simple procedure that iteratively adjusts prices and allocations in a carefully chosen, but intuitive manner. We describe a basic version of the algorithm in Section 2. The basic version already achieves a polynomial number of arithmetic operations on rationals; however, the bitlength of the rationals may be exponential in the size of the problem instance. In Section 3 we achieve polynomial time by the use of approximate arithmetic. The algorithm solves $O(n^6 \log(nU))$ flow problems and needs to deal with numbers of bitlength

Download English Version:

<https://daneshyari.com/en/article/426449>

Download Persian Version:

<https://daneshyari.com/article/426449>

[Daneshyari.com](https://daneshyari.com)