Network Flow

and

Equilibrium Computation in the Linear Exchange Economy

Kurt Mehlhorn
joint work with Ran Duan
Max Planck Institute for Informatics

June 20, 2014

Outline

Walras' Model of an Economy (Léon Walras 1875)

- each market participant (agent) owns some goods and
- has preferences over goods, i.e.,

  at a given set of prices, certain bundles of goods will give maximum pleasure (utility).

  Agents are only willing to buy bundles that give maximum utility.
- Question: are there prices such that all goods can be completely sold and agents spend all their income,i.e.

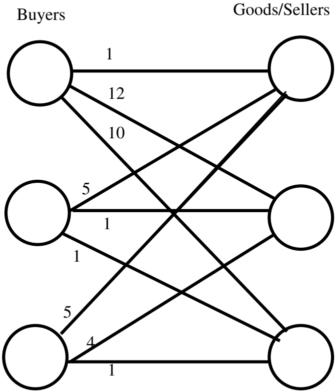
  can a perfect exchange be organized through appropriate prices?

Linear Utilities: A Special Case

- twice as much is twice as good

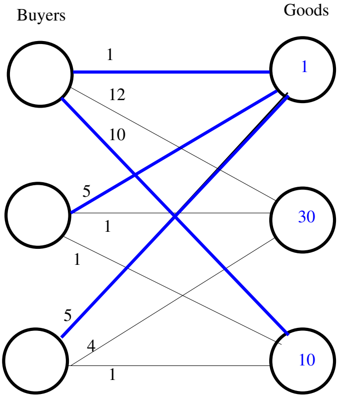
  marginal utilities do no decrease
- utilities from different goods add up
- Example: suppose a bottle of champagne gives me three times the pleasure of a bottle of wine. If the price of champagne is more than three times the price of wine, I am only willing to buy champagne, if the price is exactly three times the price of wine, I am willing to buy champagne and wine and any combination is equally fine, . . .

Example



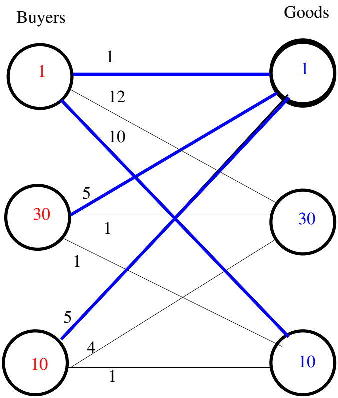
- first agent values second good 12 times as much as first good, . . .
- assume *i*-th agent owns *i*-th good, one unit of each good.

Example



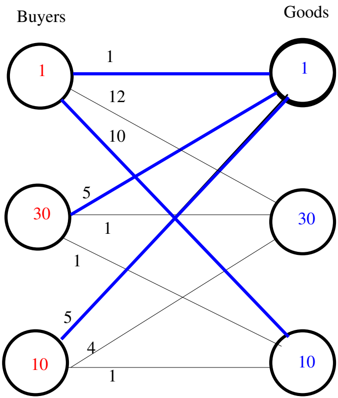
- first agent values second good 12 times as much as first good, . . .
- assume *i*-th agent owns *i*-th good, one unit of each good.
- if prices are as shown in blue, money will only flow along blue edges.

Example



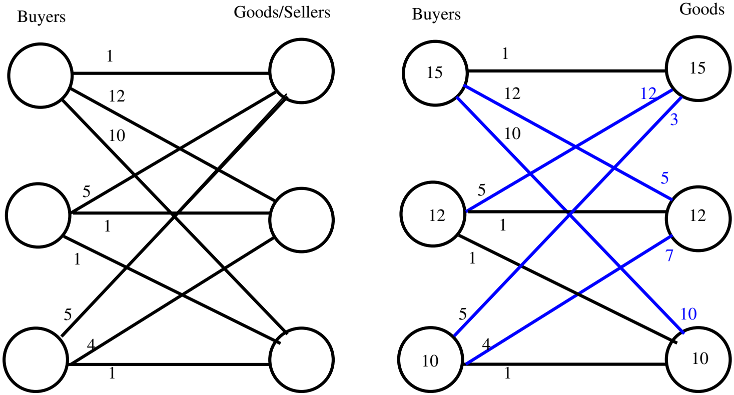
- first agent values second good 12 times as much as first good, . . .
- assume *i*-th agent owns *i*-th good, one unit of each good.
- if prices are as shown in blue, money will only flow along blue edges.
- if goods are completely sold, the red budgets will be available to the agents,

Example



- first agent values second good 12 times as much as first good, . . .
- assume *i*-th agent owns *i*-th good, one unit of each good.
- if prices are as shown in blue, money will only flow along blue edges.
- if goods are completely sold, the red budgets will be available to the agents,
- but the second good will certainly not be completely sold, because nobody is interested in it.

Example (A Solution)



utilities in black, prices inside nodes, bang-for-buck edges and flow of money in blue

The Linear Exhange Economy (Walras 1875)

- n buyers, n divisible goods — one unit of each good
- buyer i owns good i
- u_{ij} = utility for i if all of good j is allocated to him, $u_{ij} \geq 0$
- additive linear utitlities: if fraction x_{ij} of good j is allocated to buyer i, the total utility for i is

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

- p_j = price of good j — to be determined
- u_{ij}/p_j utility of good j for i per Euro
- Buyers are selfish and spend money only on goods that give them maximum utility per Euro (maximum bang per buck)
- bang per buck for buyer i: $\alpha_i = \max_j u_{ij}/p_j$

The Linear Exchange Economy

Input: Utilities $u_{ij} \geq 0$, $u_{ij} \leq U$, integral

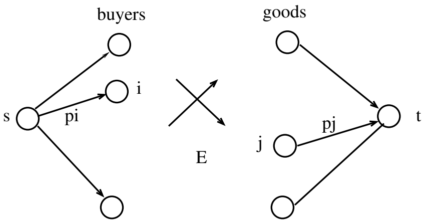
Are there prices $p_j \geq 0$, $1 \leq j \leq n$, and allocations $x_{ij} \geq 0$ such that

- all goods are completely sold: $\sum_i x_{ij} = 1$
- all money is spent: $\sum_j x_{ij} p_j = p_i$
- only bang per buck items are bought:

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

The Network G_p

Vertices: buyers b_i and goods c_j, source s and sink t



Edges:

(s, b_i) with capacity p_i

(b_i, c_j) iff $u_{ij}/p_j = \alpha_i$, unlimited capacity

(c_j, t) with capacity p_j

flow on edge (b_i, c_j) = money paid by buyer b_i for his fraction of good c_j

> p is an equilbrium iff a maximum flow saturates all edges out of s (and hence into t).

Questions

- do equilibria exist?
- properties of equilibria: is there a rational equilibrium? do equilibria form a convex set?
- algorithms:
  - approximation, exact
  - efficient
  - combinatorial or do we need ellipsoid and/or interior point
  - global knowledge versus local knowledge
  - natural updates (tatonnement)

History

Walras introduces the model in 1875 (more general utilities) and argues existence of equilibrium (iterative adaption of prices).

Fisher (1891), simpler model (buyers have budgets), alg for three buyers/goods

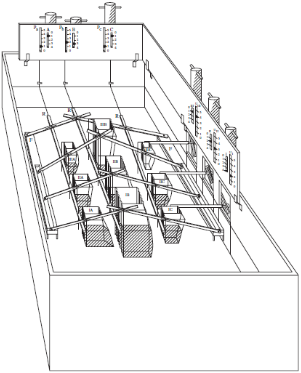
Wald (36) shows existence of equ. under strong assumptions

Arrow/Debreu (54) show existence for a much more general model under mild assumptions

Existence proofs are non-constructive (use fixed point theorems)

Figure 5

Fisher's apparatus.



Algorithm Development: Approximation Algorithms

- algorithm development starts in the 60s: Scarf, Smale, Kuhn, Todd, Eaves.
- early algorithms are inspired by fixed-point proofs or are Newton-based and compute approximations, are exponential time.
- after 2000: poly-time approximation algorithms
  - Jain/Madhian/Saberi: poly-time approximation scheme
  - Devanur/Vazirani: strongly poly-time approximation scheme
  - Garg/Kapoor: simplified approximation scheme

Exact Algorithms

- exact algorithms are based on a characterization of equilibria as the solution set of a convex program
- Nenakov/Primak (83): equilibria are precisely the solutions of

$$p_i \geq 0 \quad x_{ij} \geq 0 \quad \sum_j u_{ij} x_{ij} \geq \frac{u_{ik}}{p_k} p_i \quad \text{forall } i \text{ and } k$$

- after the substitution $p_i = e^{\pi_i}$ this becomes a convex program
- Jain (07) rediscovered this convex program and showed how to solve it with a nontrivial extension of the ellipsoid method, Ye (06) with interior point method

Combinatorial algorithms are known for the Fisher market (Devanur/Padimitriou/Saberi/Varzirani (08) and Orlin (10)); our algorithm is inspired by their work.

Our Result

Theorem (Ran Duan/KM: ICALP 2013, full paper to appear in Algorithmica)

Can compute equilibrium prices in polynomial time by a simple combinatorial algorithm.

- *alg learns about utilities by a bang-for-buck oracle.*
- *works in rounds and needs to poll the surpluses of the buyers in each round.*
- *is centralized: a central agency adjusts the prices in each round.*

Overview

intialize all prices to one: $p_j = 1$ for all j

repeat

construct the network G_p for the current prices p and compute a balanced flow f in it;

increase some prices and adjust flow;

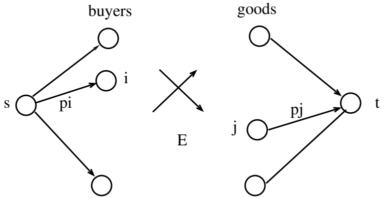
until the total surplus is tiny (less than $O(\frac{1}{4n^4 U^{3n}})$);

round the current prices to the equilibrium prices;

Details of final rounding: Let p be the current price vector; let q_i be the rational with denominator at most $(nU)^n$ closest to p_i. Then $q = (q_1, \ldots, q_n)$ is a vector of equilibrium prices.

The Flow Network G_p, Revisited

- vertices b_i, c_i, $1 \le i \le n$, s and t
- edges $E = \{(b_i, c_j) | u_{ij}/p_j = \alpha_i := \max_\ell u_{i\ell}/p_\ell\}$, capacity ∞
- let f be a maximum flow



  - $r(b_i) = p_i - \sum_j f_{ij}$, surplus of buyer i
  - $r(c_j) = p_j - \sum_i f_{ij}$, “surplus” of good j
  - $r(B) = (r(b_1), \ldots, r(b_n))$, surplus vector

- balanced flow = maxflow minimizing
$$\|r(B)\| = \sqrt{r(b_1)^2 + \ldots + r(b_n)^2};$$
- intuiton: balancing means to make surpluses more equal
- can be computed with n maxflow computations (Devanur et al)

Intuition

- Which prices should we increase?

  only prices of goods whose demand exceeds supply, i.e., goods connected in G_p to a buyer with surplus

  choose a surplus bound S, let $B(S) = \{b | r(b) \geq S\}$ and increase the prices of the goods in $C(S) =$ neighbors of $B(S)$ in G_p

- How should we increase the prices?

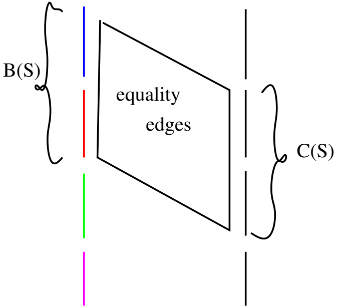
  we increase the prices of the goods in $C(S)$ by a common factor $x > 1$ and also the flows on the edges incident to the nodes in $B(S) \cup C(S)$.

- How to choose S and x?

  need to know more about the effect of changing the prices in $C(S)$ by factor x.

Price Update

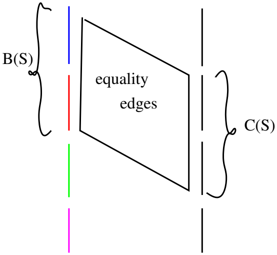
- let f be a balanced flow, order buyers $r(b_1) \geq r(b_2) \geq \ldots \geq r(b_n) \geq r(b_{n+1}) := 0$.
- let ℓ be minimal such $r(b_\ell)/r(b_{\ell+1}) \geq 1 + 1/n$, let $B(S) = \{b_1, \ldots, b_\ell\}$, and $C(S) = \{c_j | b_i \in B(S) \text{ and } (i,j) \in E\}$.



- there is no edge carrying flow from $B \setminus B(S)$ to $C(S)$
- goods in $C(S)$ have surplus zero
- increase prices of goods in $C(S)$ and flow into these vertices by a factor $x > 1$.
- surplus goes down, surplus multiplied by x, surplus goes up, surplus unchanged

Price Update, Continued

- let f be a balanced flow, let $B(S)$ be the buyers with large surplus, and $C(S)$ be their neighbors



  - goods in $C(S)$ have surplus zero
  - increase prices of goods in $C(S)$ and flow into these vertices by a factor $x > 1$.
  - surplus goes down, surplus multiplied by x, surplus goes up, surplus unchanged
  - goods in $C(S)$ keep surplus zero; goods with non-zero surplus have price one
- constraints on x
  - a new equality edge arises; goods outside $C(S)$ become more attractive for buyers in $B(S)$
  - a blue surplus becomes equal to a green or magenta surplus.
  - $x \le 1 + \frac{1}{Kn^3}$ technical reasons

The Complete Algorithm

intialize prices: $p_j = 1$ for all j

repeat

construct the network G for the current prices and compute a balanced flow f in it;

order buyers by surplus and let ℓ be minimal such that $r(b_\ell) > (1 + 1/n)r(b_{\ell+1})$. Let $B(S) = \{b_1, \ldots, b_\ell\}$.

increase prices of goods in $C(S)$ and flows into those goods by gradually increasing factor x until

new equality edge or

surplus of a buyer in $B(S)$ and a buyer in $B \setminus B(S)$ becomes equal or

$x = x_{\max} := 1 + \frac{1}{Kn^3}$ bad iteration

until the total surplus is tiny (less than $O(\frac{1}{4n^4 U^{3n}})$);

round the current prices to the equilibrium prices;

Key Lemmas

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU))$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Key Lemmas

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU))$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Key Lemmas

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU))$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Key Lemmas

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU)$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Prices stay bounded by $(nU)^{n-1}$.

order prices $p_1 \geq p_2 \geq \ldots \geq p_n = 1$ and show $p_i \leq (nU)p_{i+1}$

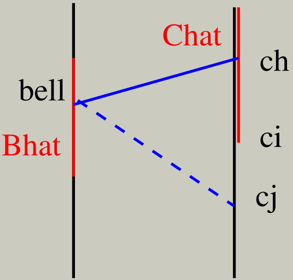
Let $\hat{C} = \{c_1, \ldots, c_i\}$, let $\hat{B}$ = buyers connected to $\hat{C}$ by E-edges.

Case 1: c_i has surplus. Then $p_i = 1$.

Prices stay bounded by $(nU)^{n-1}$.

order prices $p_1 \geq p_2 \geq \ldots \geq p_n = 1$ and show $p_i \leq (nU)p_{i+1}$

Let $\hat{C} = \{c_1, \ldots, c_i\}$, let $\hat{B}$ = buyers connected to $\hat{C}$ by E-edges.



Case 2: some $b_\ell \in \hat{B}$ likes some c_j outside $\hat{C}$, i.e., $u_{ell,j} > 0$. Let $c_h \in \hat{C}$ be connected to b_ℓ by an equality edge. Then

$$u_{\ell,h}/p_h = \alpha_\ell \geq u_{\ell,j}/p_j$$

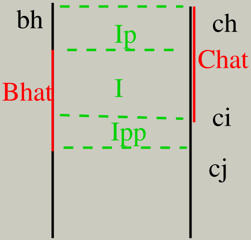
and hence

$$p_h \leq Up_j$$

Prices stay bounded by $(nU)^{n-1}$.

order prices $p_1 \geq p_2 \geq \ldots \geq p_n = 1$ and show $p_i \leq (nU)p_{i+1}$

Let $\hat{C} = \{c_1, \ldots, c_i\}$, let $\hat{B}$ = buyers connected to $\hat{C}$ by E-edges.



Case 3: $\hat{B}$-buyers like only $\hat{C}$-goods. $\hat{B}$-buyers must like a good which is not owned by one of them. Thus $I' \neq \emptyset$. Also, $p(\hat{B}) \geq p(\hat{C})$, and hence

$$p_h \leq p(I') = p(C) - p(I) \leq p(B) - p(I) = p(I''),$$

Consider $j \in I''$ with maximal p_j. Then

$$p_h \leq p(B') \leq np_j.$$

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

In each bad iteration some price increases by factor $1 + K/n^3$.

Each price can increase by this factor at most $\log_{x_{\max}}(Un)^n = n^4 \log(nU)$ times.

The Norm of the Surplus Vector

Each bad iteration increase norm by at most a factor x_{max}.

Each good iteration decreases the norm by a factor of at least x_{max}.

- choice of i: i is minimal with $r(b_{i+1}) < r(b_i)/(1 + 1/n)$.
- Thus $r(b_i) \geq r(b_1)/e \geq$ total surplus$/(en)$
- Good iteration: (1) a decreasing surplus becomes equal to an increasing or stationary surplus or (2) a new equality edge arises.
- in (2), we use new equality edge to also achieve (1)
- in (1), a surplus $\geq r(b_{i+1})$ and a surplus $\leq r(b_i)$ becomes equal.

The Norm of the Surplus Vector

Each bad iteration increase norm by at most a factor x_{max}.

Each good iteration decreases the norm by a factor of at least x_{max}.

Number of good iterations is $O(n^5(\log(nU))$.

This many iterations to make up for the bad iterations.

Similar number of iterations to make the total surplus tiny.

Summary

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU))$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Summary

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU))$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Summary

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU))$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Summary

Prices stay bounded by $(nU)^{n-1}$.

Number of bad iterations is $O(n^5 \log(nU))$.

Norm of surplus vector decreases by factor $1 + \Omega(1/n^3)$ in good iterations and increases by factor $1 + O(1/n^3)$ in bad iterations.

Number of good iterations is $O(n^5 \log(nU))$.

It suffices to compute with number with $O(n \log(nU))$ bits.

Running time = $O(n^5 \log(nU) \cdot n \cdot n^3 \cdot n \log(nU))$.

Open Problems

More complex utility functions

Huge amount of work on approximation algorithms by many: Vijai Vazirani, Kamal Jain, Jugal Garg, Nikhil Devanur, Christos Papadimitriou, Ruhta Mehta, ...

Strongly polynomial algorithms

James Orlin (2011): strongly polynomial alg for Fischer model.

Ongoing markets and/or local algorithms

very interesting work by Yun Kuen Cheung, Richard Cole, Lisa Fleischer, and Ashish Rastogi