

Computing Market Equilibria

Part II – non-linear utilities

Dariusz Kowalski

MPI fuer Informatik, Germany

and

Uniwersytet Warszawski, Poland

Outline

Polynomial algorithms computing (approximate) equilibria

- Extended model: spending constraint

Utilities:

- depend on prices p ,
- defined for each agent and each good (also for money),
- concave, piece-wise linear functions

[Devanur, Vazirani STOC'2004] [Vazirani 2004]

- Basic Fisher model, Leontief utilities (with extension to more general CES utilities)

[Codenotti, Varadarajan ICALP'2004]

[Codenotti, Jain, Varadarajan, Vazirani 2004]

Spending Constraint Model

A market with set A of agents and set G of goods. Assume $|A| = n$. We assume that money are also a kind of a good, denoted by 0, but for technical reason $0 \notin G$.

Agent a maintains

- Amount (integral) m_a of *money*,
- *Rate functions*, for each good g including money,

$$f_{a,g} : [0, m_a] \rightarrow \mathbf{R}_+$$

Function $f_{a,0}$ specifies the rate at which agent a derives utility per dollar as a function of the amount she does not spend.

Spending Constraint Utilities

If *prices* p_g , for $g \neq 0$, are fixed, we define *utility functions*

$$u_{a,g} : [0, m_a] \rightarrow \mathbf{R}_+ \text{ as follows } u_{a,g} = \int_0^x \frac{f_{a,g}(y)}{p_g} dy$$

Additionally, utility function for money $u_{a,0} : [0, m_a] \rightarrow \mathbf{R}_+$ is defined

$$u_{a,0} = \int_0^x f_{a,0}(y) dy$$

If $f_{a,g}$ is *continuous* and *monotonically decreasing*

then $u_{a,g}$ will be *strictly concave* and *differentiable*.

Conclusion: For fixed price vector \mathbf{p} each function $u_{a,g}$ has a *unique* allocation maximizing this function.

Objective: Buy goods (and keep money) that maximize all utilities

Market clearing prices

Goal:

Find a price vector \mathbf{p} (*equilibrium prices*) such that:
if each agent selects its optimal basket of goods
(including money which she doesn't want to spend)
the market clears (no surplus, no deficit).

Algorithm in time polynomial in n , $|G|$, $\log \sup_{a,g} f_{a,g}$
and $\log \sum_{a \in A} m_a$

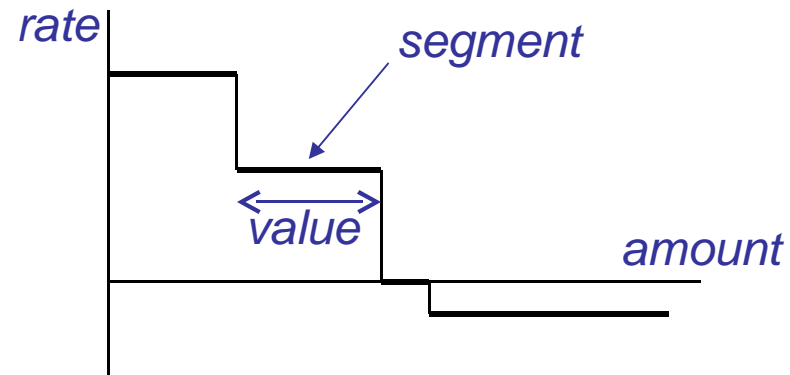
Step-decreasing (rate) functions

Let f be a step-decreasing rate function. We say that $s = [y, z]$ is a *segment* of f iff there is c such that

- for every $v \in s$, $f(v) = c$, and
- for every $v \notin s$, $f(v) \neq c$.

We say that

- $(z - y)$ is a *value* of segment s ,
- c is a *rate* of segment s ,
- g is a *good* (including money) of segment s .



Segment s is called a *segment of agent a* iff it is a segment of rate function $f_{a,g}$, for some good g (including money).

Existence and Uniqueness

Theorem [Devanur, Vazirani 2004]

Consider a market where $f_{a,g}$ are **step-decreasing functions**, satisfying the following conditions:

- For each good g there is a *potential buyer*: there is an agent a and segment s of $f_{a,g}$ such that rate of this segment is positive;
- Each agent wants to *use all her money* (to buy or to keep some money unspent): the sum of values of positive-rate segments of agent a is at least m_a .

For this market there exist **unique** clearing prices.

Computational Results

- Polynomial algorithm for finding equilibrium prices
- Algorithm based on the algorithm by Devanur, Papadimitriou, Saberi and Vazirani [2002] for linear utilities, presented before – we will call it **algorithm DPSV**

Overview of the Algorithm

- Initialize:
 - small starting prices $1/n$,
 - defining starting equality subgraph, with frozen empty sets of agents A_1 and goods G_1 , and active sets of agents $A_2 = A$ and goods $G_2 = G$
 - if some good is disconnected from set of agents, reduce its price to make it attractive for some agent (reduce to the biggest such price)
- While G_2 – set of active goods – is not empty
 - increase active prices continuously, scaled by some factor, until network will **change** or there will be new **tight set** (one of **four** events may happen)
 - perform action connected with this event:
modify: network, frozen and active sets, allocation of goods

Similarities

- **Tight sets** are crucial – goods in such set have market-clearing prices
- Procedure Freeze – finding **maximal tight set** and freeze it, using min-cut in current network, then allocating goods in frozen component (in optimal way)
- Finding **new tight set** – to determine when to stop increasing prices to satisfy Event 1 or Event 4
- Modification of the algorithm, via **balanced flows**, to obtain polynomial time algorithm (described basic solution may not guarantee a polynomial time)

Differences

Utility functions are not linear and depend on prices

Consequences:

- Modified definition of **equality networks**, based on **segments** instead of goods
(but still we use fixed set of parameters to do it)
- When prices are increasing, some **active good** may become attractive to some **frozen agent**, since higher prices decrease amount of goods which may be allocated to the agent – hence now goods may have higher rate (concave) – **Event 3**

Money are treated as a good

Consequences:

- Additional **Event 4** may occur

Differences – Equality Network

- For each segment $s \in f_{a,g}$, which is among most preferable active segments for agent a , we connect a with g
- Preference is measured according to values $rate(s)/price(g)$

For linear utilities instead of segments s we used value $u_{a,g}/price(g)$ from utility function u_a

- Capacity of edge (g,a) is set to the $value(s)$
- Capacity of an edge from the source to a good g is a current price of g minus $allocated(g)$, and capacity of an edge from an agent a to the sink is a current amount of money of agent a not spent for already allocated goods (including kept money)

For linear utilities we did not consider money as a good, and did not consider $allocated(g)$ while setting a capacity of edge to g

Consequences

Procedure Freeze finding a maximum tight set slightly changes.

But still we have to find min-cut and modify it recursively.

Events

Events are to maintain the following invariant during the process of increasing prices of active goods:

$$\text{For every } G' \subseteq G, m(G') \leq m(\Gamma(G))$$

which is the same as in the original algorithm DPSV.

Hence the skeleton of correctness and complexity analysis' are similar to algorithm DPSV.

This invariant may be unsatisfied if:

- Some set of active goods become tight, or
- Topology of the network changes (some new edges are added)

Event 1

Some set $G^* \subseteq G_2$ become tight.

This is good since set of incident agents is optimally served.

Action:

Perform procedure Freeze finding maximum such set

When to stop increasing prices?

The same procedure as in the algorithm DPSV

Event 2

- Prices of frozen goods are decreasing comparing to the prices of active goods
- Some frozen good g becomes most preferable for some active agent a (only allocation of active goods changes dynamically)
- Agent a becomes adjacent to good g – **network changes!**

Let s be the segment in $f_{a,g}$ corresponding to this event (most preferable for agent a)

Action:

Add edge (g,a) with capacity $value(s)$ to the network and perform Freeze

When to stop increasing prices?

Examine all segments – for each compute the first time when it may cause such event

Event 3

- Prices of active goods are increasing
- Amounts of active goods allocated to frozen agents are decreasing
- Some frozen agent a may start to allocate active goods using higher rate segment of utility functions, say segment s of $f_{a,g}$
- Attractiveness $rate(s)/price(s)$ of the active good g may increase, and it may become most attractive for the frozen agent a
- Frozen agent a becomes adjacent to the active good g – network changes!

Action:

- Deallocate $value(s)$: subtract it from $allocated(g)$ and $spent(a)$
- Since $m(a)$ increases, G_1 is not tight anymore; hence call Freeze

When to stop increasing prices?

Examine all segments – for each compute the first time when it may cause such event

Event 4

Money become most preferable for some agent, and still there are some remaining to ``keep''

Action:

Continuously increase the amount of ``kept'' money until either all money are ``kept'' or some set of goods becomes tight and perform Freeze

When to stop increasing prices?

Find when some set may become tight (similarly to Event 1)

Correctness and Complexity

Argument similar to one from analysis of algorithm DPSV:

- We are increasing prices and freezing tight sets of goods and their incident agents, since they are perfectly balanced;
- When total increase is $\sum_{a \in A} m_a$ algorithm finishes.

Complexity: polynomial in $\left(\sup_{a,g} f_{a,g} \right)^{|G|}$
Reduce time to polynomial in $n, |G|, S, \log \sup f_{a,g}$ and $\log \sum m_a$ using the same method as DPSV, based on balanced flows

Nice Rate Functions

Rate functions are *nice* iff

- Each $f_{a,g}$ is either **continuous** and **strictly decreasing**, or **identically zero**;
- For every agent a there is a **non-zero function** $f_{a,g}$.

Theorem [Devanur, Vazirani 2004]

If rate functions are **nice** then price's and allocation's equilibrium vectors are **unique**.

Remarks

- For nice functions we have PTAS algorithm computing ε -approximate equilibrium.

A price vector p is an ε -approximate equilibrium if neither deficiency nor surplus of goods is too high:

$$\sum_{g \in G} |\xi_g - p_g| \leq \varepsilon \sum_{g \in G} p_g$$

where ξ_g denotes how much money agents want to spend on good g given price vector p .

Solution: Approximate given rate function by step-decreasing rate function, where each segment has length ε . Call previous algorithm.

Basic model with non-linear utilities

Go back to the basic Fisher's model:

- Money are not treated as a good
- There is one utility function per agent and it does not depend on prices

Fisher's Model with CES Utilities

Let a matrix $\alpha = (\alpha_{a,g})_{a \in A, g \in G}$ be fixed, and

let σ be a constant, called *elasticity of substitution*.

- CES (*constant elasticity of substitution*) utility functions

$$u_a(x_a) = \left(\sum_{g \in G} \alpha_{a,g}^{1/\sigma} x_{a,g}^{1-1/\sigma} \right)^{\frac{\sigma}{\sigma-1}}$$

- Leontief utility function (CES for $\sigma \rightarrow 0$)

$$u_a(x_a) = \min_{g \in G} \frac{x_{a,g}}{\alpha_{a,g}}$$

- Cobb-Dauglas utility function, where $\alpha_{a,g}$ are such that

$$\alpha_{a,1} + \dots + \alpha_{a,|G|} = 1, \text{ for every agent } a \quad (\text{CES for } \sigma \rightarrow 1)$$

$$u_a(x_a) = \prod_{g \in G} x_{a,g}^{\alpha_{a,g}}$$

Computing the Equilibrium

Simpler algorithm for Leontief utilities:

Step 1: Solve some **convex optimization problem**, denoted by **CP(β)**, corresponding to considered market, and obtain argument β^* maximizing the given function.

Step 2: Solve another **linear algebra problem**, denoted by **FP(p)**, to obtain a partial price vector p^* solving the given equations.

Step 3: Extend vector p^* by zero's to obtain price vector p^* clearing the market, and compute best baskets for agents.

Step 1

Let $\mathbf{1}$ denote a vector $(1, \dots, 1) \in \mathbf{R}^{|G|}$. Let $m = m_1 + \dots + m_n$.

Problem CP(β):

$$\begin{aligned} & \text{maximize} && \left(\prod_{a \in A} \beta_a^{m_a} \right)^{1/m} \\ & \text{subject to} && \alpha \leq \mathbf{1} \\ & && \beta_a \geq 0, \text{ for every } a \in A \end{aligned}$$

We solve it in polynomial time using e.g., ellipsoid method for solving convex optimization problems.

We obtain argument β^* maximizing the given function.

Step 2

Let $G^t = \{g: \alpha_{1,g} \beta_1^* + \dots + \alpha_{n,g} \beta_n^* = 1\}$ be a set of goods for which the corresponding constraint is tight, and let G^l denote the remaining goods.

Problem FP(p):

$$\begin{array}{ll} \text{compute} & p \\ \text{subject to} & \sum_{g \in G^t} p_g \beta_a^* \alpha_{a,g} = m_a \text{ for every } a \in A \\ & p_g \geq 0 \text{ for every } g \in G^t \end{array}$$

We solve it in polynomial time using e.g., ellipsoid method for solving convex optimization problems.

We obtain argument p^* solving the given equations.

Step 2 (*cont.*)

Claim

Problem $\text{FP}(p)$ has a solution p^* .

Proof:

If it is not true, using Farkas Lemma we obtain vector (t_1, \dots, t_n) such that $t_1 m_1 + \dots + t_n m_n > 0$, which yields contradiction with the fact that β^* is a solution of $\text{CP}(\beta)$ problem.

Step 3

- Extend computed partial vector p^* to price equilibrium \mathbf{p}^* by setting $p_g^* = 0$ for each $g \in G^l$
- Set the equilibrium baskets x_a^* of an agent a as follows

$$x_{a,g}^* = \beta_a^* \alpha_{a,g} \quad \text{for each } a \in A \text{ and } g \in G^t$$

$$x_{a,g}^* = \beta_a^* \alpha_{a,g} \quad \text{for each } a \neq 1 \text{ and } g \in G^l$$

$$x_{1,g}^* = 1 - \sum_{a \neq 1} x_{a,g}^* = 1 - \sum_{a \neq 1} \beta_a^* \alpha_{a,g} \quad \text{for each } g \in G^l$$

Correctness and Complexity

Theorem [Codonotti, Varadarajan 2004]

Let \mathbf{p}^* and x^* be the corresponding equilibrium prices and equilibrium basket of goods. For every agent a , we define $\beta'_a = u_a(x_a^*) = \min_{g \in G} \frac{x_{a,g}}{\alpha_{a,g}}$. Then β' is an **optimal** solution for CP(β).

Corollary

Since objective function of CP problem is strictly quasi-concave, β' is a **unique optimal** solution for CP(β).

Asymptotic time complexity:

$$O(\text{time of ellipsoid algorithm} + n |G|)$$

Unique and Irrational Solutions

Theorem [Codenotti, Varadarajan 2004]

The equilibrium basket of goods is **unique**.

(Equilibrium prices **need not be unique**!)

Example: consider two goods and three agents with

$m_1 = m_2 = m_3 = 1$, and $\alpha_{1,1} = 1/3$, $\alpha_{2,1} = 1/6$,

$\alpha_{3,1} = 1/12$, $\alpha_{1,2} = 1/6$, $\alpha_{2,2} = 1/3$, $\alpha_{3,2} = 1/15$.

Solution of CP(β) gives $\beta^* = (2/\sqrt{3}, 1 + 1/\sqrt{3}, 10 - 10/\sqrt{3})$.

Consequently, prices and baskets must contain **irrational numbers**.

Remarks

- Equilibria can be irrational – hence we need to round our solution, which gives PTAS algorithm finding ϵ -approximate equilibrium for such irrational solutions.
- For Codd-Dauglas utilities, Eaves [1985] designed polynomial time algorithm finding equilibrium prices and equilibrium basket
- Described method can be extended to finding equilibria for CES utilities [Codonotti, Jain, Varadarajan, Vazirani 2004]

Conclusions

- Polynomial algorithm computing equilibria for market with spending constraint utilities – for piece-wise linear utilities which depend on prices
- Polynomial algorithm for computing equilibria in basic model with Leontief, or more general with CES, utilities, if such utilities are rational; otherwise we round a solution and obtain approximate equilibria in polynomial time

Another Open Problems

- Existing of polynomial time algorithms for considered two models and more general (sub-)class of concave utility functions.